LEVEL II

1

FLOW RESEARCH COMPANY

A DIVISION OF FLOW INDUSTRIES, INC.

AD A067795

DDC FILE COPY

DDC
APR 24 1979
C

This document has been approved for public release and sale; its distribution is unlimited.

HEADQUARTERS
21414 - [illegible] Avenue South
Kent, Washington [illegible] 1370
[illegible] TWX [illegible]

DISCLAIMER NOTICE

12

159p.

9

Flow Research Note No. 95

6

Turbulent Flow Past a Self-Propelled Vehicle II. Computation

14 FLOW-RN-95

10

by

Walter J. Grabowski [redacted] Robert E. Robins

December 1976 (Revised December 1977 and November 1978)

11

Sponsored by:

Defense Advanced Research Projects Agency
ARPA Order No. 1910

15

Contract No. N00014-76-C-0564, ARPA Order-1910
Program Code No. NR062-541/12-11-75
Name of Contractor: Flow Research Company
A Division of Flow Industries, Inc.
Principal Investigator and Phone Number: Dr. James J. Riley
(206) 854-1370

Scientific Officer: R. Mindak, Office of Naval Research
Effective Date of Contract: 1 February 1976
Contract Expiration Date: 31 October 1977
Amount of Contract: $107,940.00
Short Title of Work: Turbulent Wake Flow

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.

390 409

Flow Research Note No. 95
November 1978

ABSTRACT

The ICWAKE computer code solves the Navier-Stokes equations for axisymmetric, incompressible, swirling, turbulent flow with large axial gradients. This document is a guide to the use of the code. Included are descriptions of the input parameters and the code structure, some general comments about using the code and a sample calculation.

TABLE OF CONTENTS

		Page
	Abstract	i
	Table of Contents	ii
1.	Introduction	1
2.	Description of Input Quantities	2
3.	ICWAKE Code Structure	13
4.	Use of ICWAKE -- General Observations	30
5.	Sample Calculation	32
	References	68
	Appendix	69

1. INTRODUCTION

This manual describes the structure of the ICWAKE computer code and the use of this code to solve the Navier-Stokes equations for axisymmetric, incompressible, swirling, turbulent flow with large axial gradients. The mathematical analysis and finite-difference formulation upon which it is based are described in the companion document "ICWAKE COMPUTER CODE —— Mathematical Analysis and Finite-Difference Formulation" (Grabowski, et al. 1976). Prospective users of the code should be familiar with this reference. Experience in the use of large hydrodynamics computer codes is necessarily assumed because the code obtains a finite-difference solution for an elliptic system of eleven rather lengthy, coupled, partial differential equations.

ICWAKE, which is written in FØRTRAN IV, was developed and used extensively on the Lawrence Berkeley Laboratory CDC 7600 computing system. It can be compiled to produce executable object codes on the LBL RUN76, FTN and FTN4 compilers. ICWAKE requires approximately 160_8K storage to execute with a maximum grid system of 60 x 32 points. It makes use of the NCAR direct Poisson solver, BLKTRI.

ICWAKE is based on a code, FINDØM, for laminar flow, which was written under the sponsorship of NASA/Ames Grant NGR 05-003-451 at the University of California, Berkeley. Further development and the extension to turbulent flow were performed under the sponsorship of a Flow Research, Inc., Independent Research and Development Program, and most recently under the sponsorship of ONR Contract No. N00014-76-C-0564.

In section 2 of this manual, we describe the required input quantities for code execution. In section 3, we outline the structure of the code and the roles of the main routine and the subroutines. In section 4, we present some general observations about the use of the code and the choice of parameters. Finally, in section 5 we present input and output for a complete calculation.

2. DESCRIPTION OF INPUT QUANTITIES

The input data necessary to exercise ICWAKE enter the code through Namelists DAT1 through DAT6, DIS7, and DAT8, which are read by the main program, subroutine DATIN, and subroutine STRESS2, respectively. All input parameters for which no dimensions are specified are dimensionless. Data from previous calculations can be entered through logical unit 41 (TAPE41), which is read by ICWAKE, and can be used as the starting conditions. (Using data from previous calculations as initial conditions is discussed in the description of routines UPCØND, DATIN, and IUNI in Section 3.)

Namelist $DAT1

$DAT1 is read at the beginning of each exercise of the code. It provides the following: the parameters that determine the frequency and scope of the printout; the time-step size, and a maximum number of time-steps to be taken in the computation; a convergence criterion, and the frequency at which convergence checks are performed; the sequence number of the computer run, and the time-step count at the beginning of the run.

$DAT1

M1,M2,M3	Output printout control at intermediate time-steps. Solution is printed out for all I (i.e., x) at values of J from M1 through M2 in increments of M3 (Default values = 1,1,1)
M11,M22,M33	Output printout control at termination of the run (Default values = 1,8,1)
NPRNT	Frequency (in terms of time-steps) of intermediate printout (Default value = 30)
TA	Time-step size (Default value = 0.1)
NTMX	Maximum number of time-steps to be taken (Default value = 30)
CØNCRIT	Convergence criterion based on the time rate of change of the rms divergence of mean velocity (Default value = 1.E-9)
NTCHK	Frequency (in terms of time-steps) of convergence checks (Default value = 10)
NUMBER	Sequence number of the run (Default value = 1)
NSTRT	Time-step count at the beginning of the run (Default value = 0)

Namelist $DAT2

$DAT2 is read at the beginning of each exercise of the code in which data from previous calculations is not used as input (i.e., TAPE41 not read). It contains the parameters necessary to set up the computational grid system and to generate the coefficients of the logarithmic transformation. Generating these four coefficients requires specifying the total number of grid points desired in each direction, as well as the number of points desired between ZINITL and some distance ZCC (less than ZMAX), and between the axis of symmetry and some distance RCC (less than RMAX).

$DAT2

N	Total number of grid points in the x (or z) direction (Default value = 40)
M	Total number of grid points in the y (or r) direction. M must equal 2^k, where k is an integer greater than one (Default value = 16)
NC	Number of grid points in the x direction required between ZINITL and ZCC (Default value = 10)
MC	Number of grid points in the y direction required between the axis (r = 0) and RCC (Default value = 8)
ZINITL	The z location of the WEST (upstream) boundary. This will correspond to x = 0 (Default value = 0)
ZCC	A length in the z direction beginning at z = 0. NC points are to be located between ZINITL and ZCC. (ZCC must be greater than ZINITL) (Default value = 1)
RCC	A length in the r direction beginning at the axis, in which MC points are to be located (Default value = 1)
ZMAX	z location of the EAST (outflow boundary) (at ZMAX, x = 1) (Default value = 20)
RMAX	r location of the NORTH (radial) boundary (Default value = 6)
YMAX	y value assigned to the radial boundary RMAX (Default value = 0.5)
AX,AY	Initial estimates of the axial and radial logarithmic transformation parameters (Default values = 3.076 and 5.8, which correspond to the default grid system)
EPS	Convergence coefficient for the logarithmic transformation coefficient calculation procedure (Default value = 1.E-6)

MTURB — The NORTH boundary conditions on all turbulence quantities are applied at J = MTURB (Default value = M)

Namelist $DAT3

$DAT3 provides the flow Reynolds number, a laminar-turbulent flag, a flag that may be set to decouple the mean flow calculation from the turbulence calculation, and a flag that may be set to apply Mager's (1972) cubic-quartic profiles at the WEST boudnary. Note that $DAT3 is not read during calculations with NSTRT greater than zero.

$DAT3

RE	Flow Reynolds number based on the assumed characteristic velocity and length
IDECØUP	A flag that, if set to unity, will cause the mean flow calculations to be decoupled from the turbulence calculation (Default value = 0)
ITURB	A flag that must be set to either zero or unity for laminar or turbulent flow, respectively (No default value)
IMAGER	A flag that is set to unity in order to apply Mager's cubic-quartic upstream profiles for a laminar calculation (Default value = 0)
WI	Free-stream axial velocity for Mager's quartic profile
VI	Free-stream circulation for Mager's cubic profile
ALPH	Form factor for Mager's quartic profile

Note that WI, VI, and ALPH have to be set only when IMAGER = 1. They have no default values.

The following parameters are required only when IBLPRP = 1 (see namelist $DAT4); they should be given in CGS units. (No default values are assigned.)

BRD	Body radius
BLN	Body length
BUIN	Body velocity (free-stream axial velocity)
RPRØP	Propeller radius (distance from the axis of revolution to the propeller tip)
ALAMB	Advance ratio of the propeller, λ = BUIN/(RPRØP*Ω), where Ω is the angular velocity of the propeller (rad/sec)
NB	Number of blades on the propeller

NBEL	Number of blade elements
BELR	An array containing the radius of each blade element
BELC	An array containing the chord length of each blade element
BELANG	An array containing the geometric pitch angle (in degrees) of each blade element
BELTH	An array containing the maximum thickness of each blade element
BELDEL	An array containing the maximum displacement of the mean camber line from a chord line connecting the leading and trailing edges

Namelist $DAT4

$DAT4 provides radial profiles of the mean flow and turbulence quantities at z = ZINITL (or correspondingly, x = 0). $DAT4 is not read when IMAGER = 1. If a laminar flow (but not Mager's profiles) is to be computed (i.e., IMAGER = 0, ITURB = 0), only MPØINT, RT, UL, VL, and WL must be specified. If a turbulent flow is to be computed and IBLPRP = 0 (see below), the $DAT4 arrays are needed. If some are not available, subroutine DATIN should be modified to compute or estimate them.

$DAT4

MPØINT	The number of data points at the WEST boundary
RT	WEST boundary data point location array (MPØINT values) (Note that RT(1) must equal zero and that when $DAT4 is read, RMAX is set to RT(MPØINT). Also, note that all of the arrays below consist of MPØINT elements defined at the RT data points).
UL	Array of radial velocities, U.
VL	Array of circumferential velocities, V.
WL	Array of axial velocities, W.
UU	Array of radial-radial velocity fluctuation correlations, R_{rr}.
VV	Array of circumferential-circumferential velocity fluctuation correlations, $R_{\theta\theta}$.
WW	Array of axial-axial velocity fluctuation correlations, R_{zz}.
WU	Array of axial-radial velocity fluctuation correlations, R_{zr}.
WV	Array of axial-circumferential velocity fluctuation correlations, $R_{z\theta}$.
UV	Array of radial-circumferential velocity fluctuation correlations, $R_{r\theta}$.

(No default values are assigned to the above arrays.)

ISCHETZ	When this signal is set at 1, the $R_{r\theta}$ turbulence correlation is initialized according to an eddy viscosity formulation (Default value = 0)
IDEP	When this signal is set at 1, the dissipation rate ε is initialized by setting it equal to the production rate. If IDEP = 0, then $\varepsilon = K_\varepsilon k^{3/2}/\ell$, where $k = \frac{1}{2}(R_{rr} + R_{\theta\theta} + R_{zz})$, and K_ε and ℓ are assigned through namelist $DIS7. (Default value = 0)

IBLPRP — When this signal is set at 1, all variables are initialized according to the boundary-layer/propeller algorithm contained in subroutines BLAYER, PRØPWV, PRØPU, and PTURB (see namelist $DAT3). (Default value = 0)

Namelist $DAT5

$DAT5 provides main program iteration-sweep control. Instead of the ADI procedure, either horizontal or vertical line-by-line iteration may be specified. (The default ADI procedure is strongly recommended.)

$DAT5

ISWEEPX	A flag that should be set to zero to skip horizontal ADI sweep (Default value = 1)
ISWEEPY	A flag that should be set to zero to skip vertical ADI sweep (Default value = 1)

Note that the computation will abort if both ISWEEPX and ISWEEPY equal zero.

Namelist $DAT6

$DAT6 provides the axial convection weighting factors as a function of axial location.

$DAT6

ARTVIS — An array (N values) of axial convection weighting factors as x (or z) locations. ARTVIS may be set to zero, unity, or some function between, for purely central, purely upwind, or combined differencing, respectively. ARTVIS should always be set to unity at x grid points N and N-1, and should vary smoothly to these values.

Namelist $DIS7

$DIS7

TSCALE — *The value ℓ in the dissipation rate equation,* $\varepsilon = K_\varepsilon k^{3/2}/\ell$ (see IDEP in namelist $DAT4). (Default value = 0.2)

XKECØN — The value K_ε in the above equation for the dissipation rate (Default value = 0.53)

Namelist $DAT8

$DAT8 is read by STRESS2. It provides turbulence constants and also sets flags that determine the form of the boundary condition applied to the circumferential-circumferential velocity fluctuation correlation at the axis, in addition to flags that determine the particular turbulent diffusion model to be used.

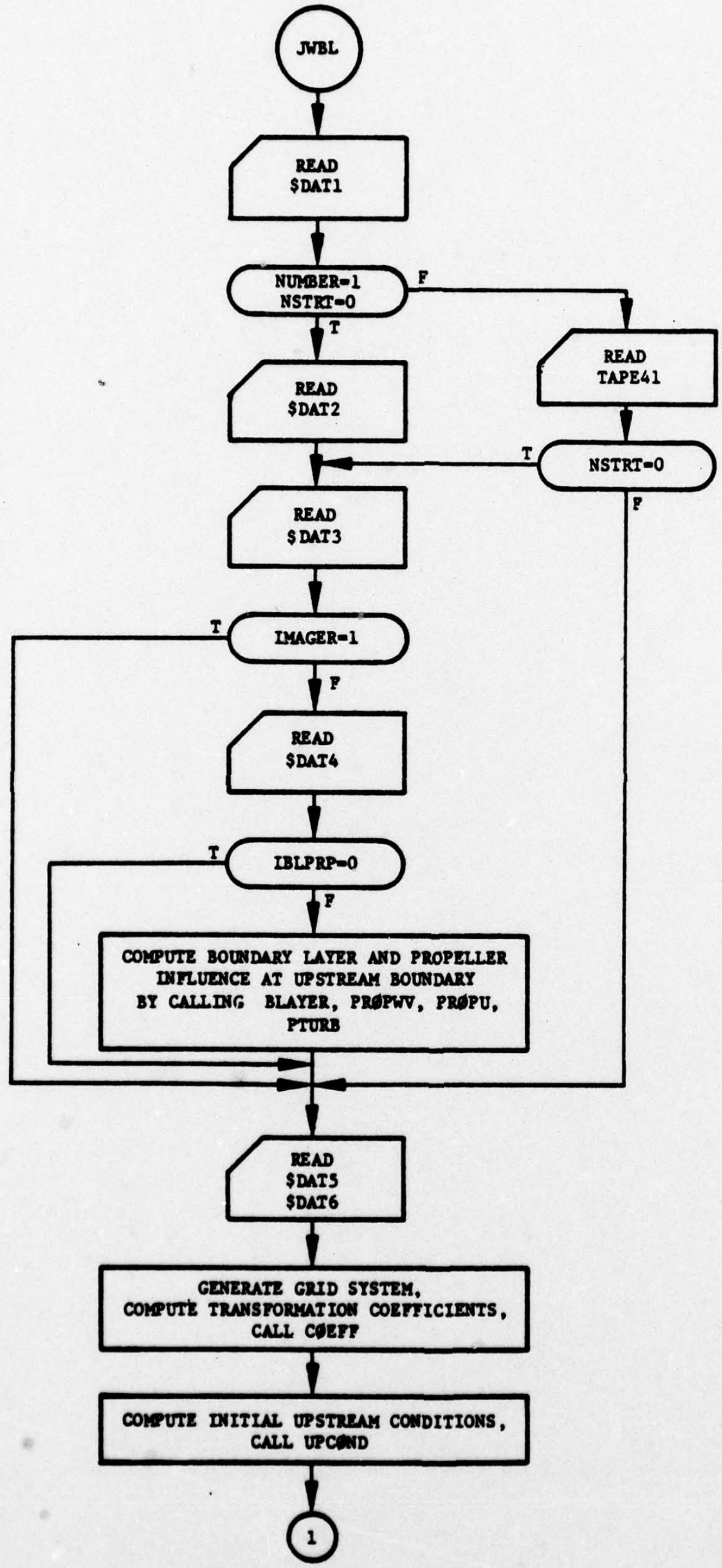
$DAT8

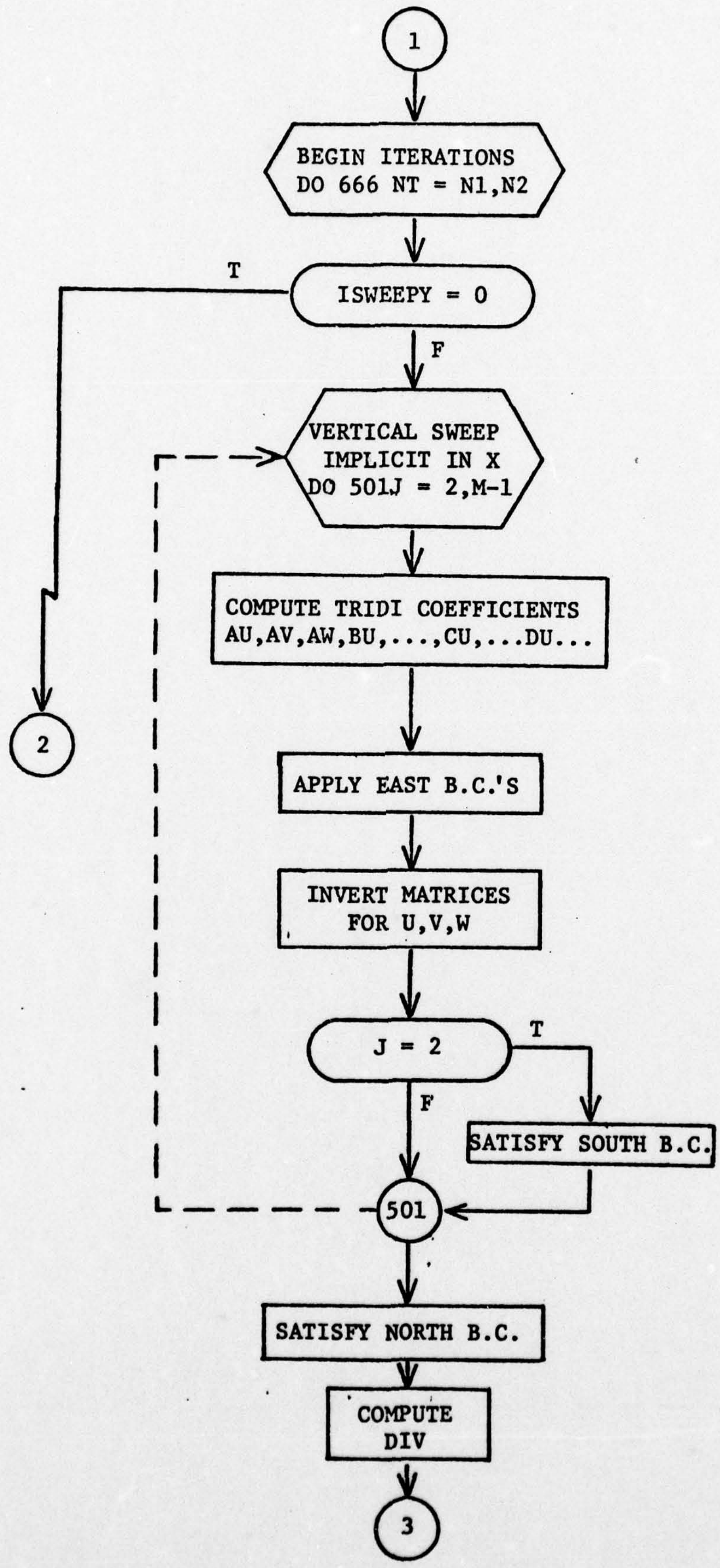
NHL	A flag set to zero or unity for Daly-Harlow or Hanjolic-Launder turbulent diffusion, respectively.
CEPS	Turbulence model constant (Default value = 0.15)
CEPS1	Turbulence model constant (Default value = 1.44)
CEPS2	Turbulence model constant (Default value = 1.90)
CSO	Turbulence model constant (Default value = 0.25)
CS1	Turbulence model constant (Default value = 0.11)
CSN	Turbulence model constant equal to CSO if NHL = 0, and CSN if NHL = 1
CØN1	Turbulence model constant (Default value = 1.5)
CØN2	Turbulence model constant (Default value = 0.4)
ISBCTT	Boundary condition control at R = 0. Setting ISBCTT = 0 requires $\partial R_{rr}/\partial r = 0$, $R_{\theta\theta} = R_{rr}$ at $r = 0$, while setting ISBCTT = 1 requires $\partial R_{rr}/\partial r = \partial R_{\theta\theta}/\partial r = 0$ at $r = 0$ (Default value = 1)

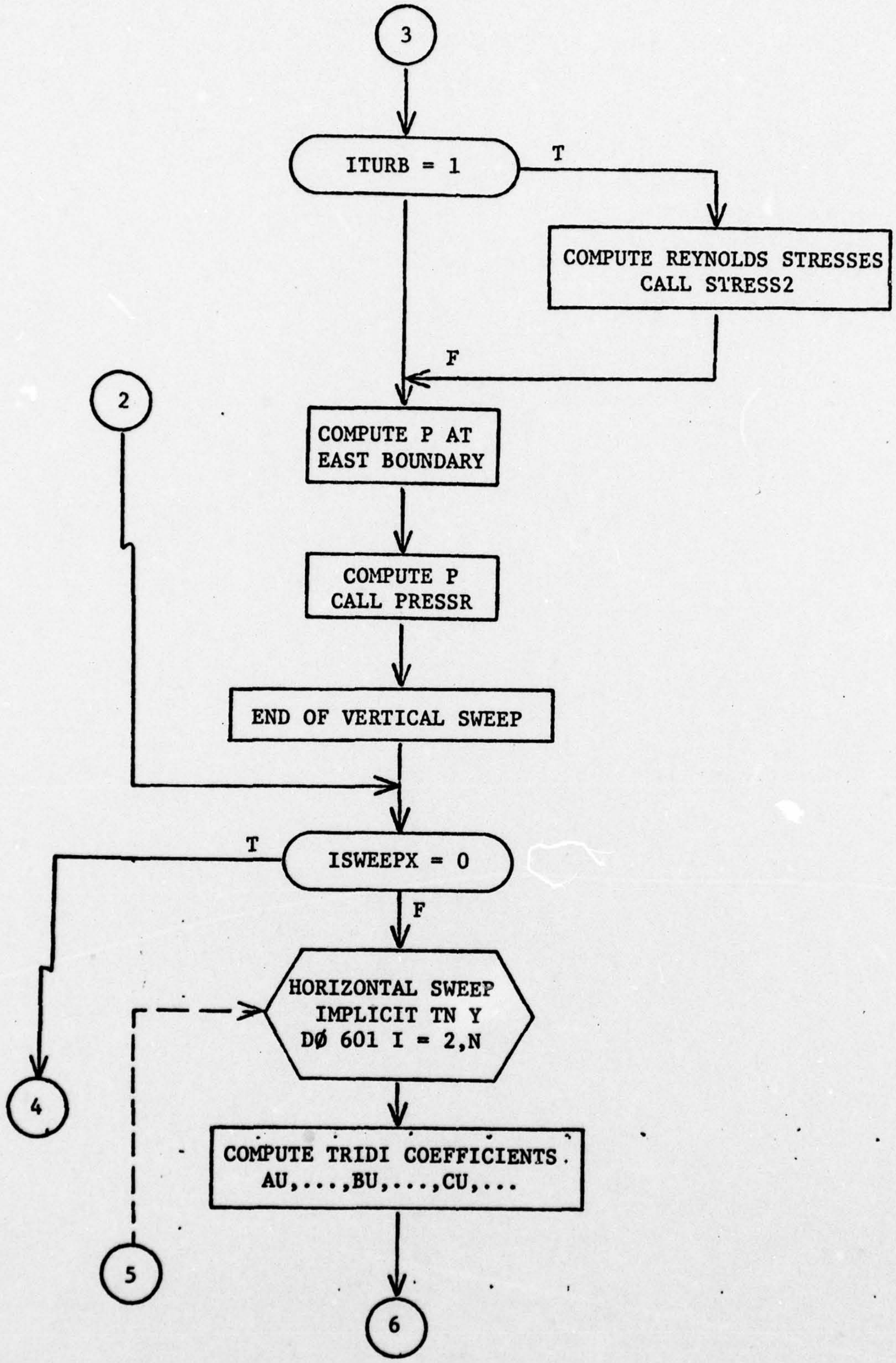
3. ICWAKE CODE STRUCTURE

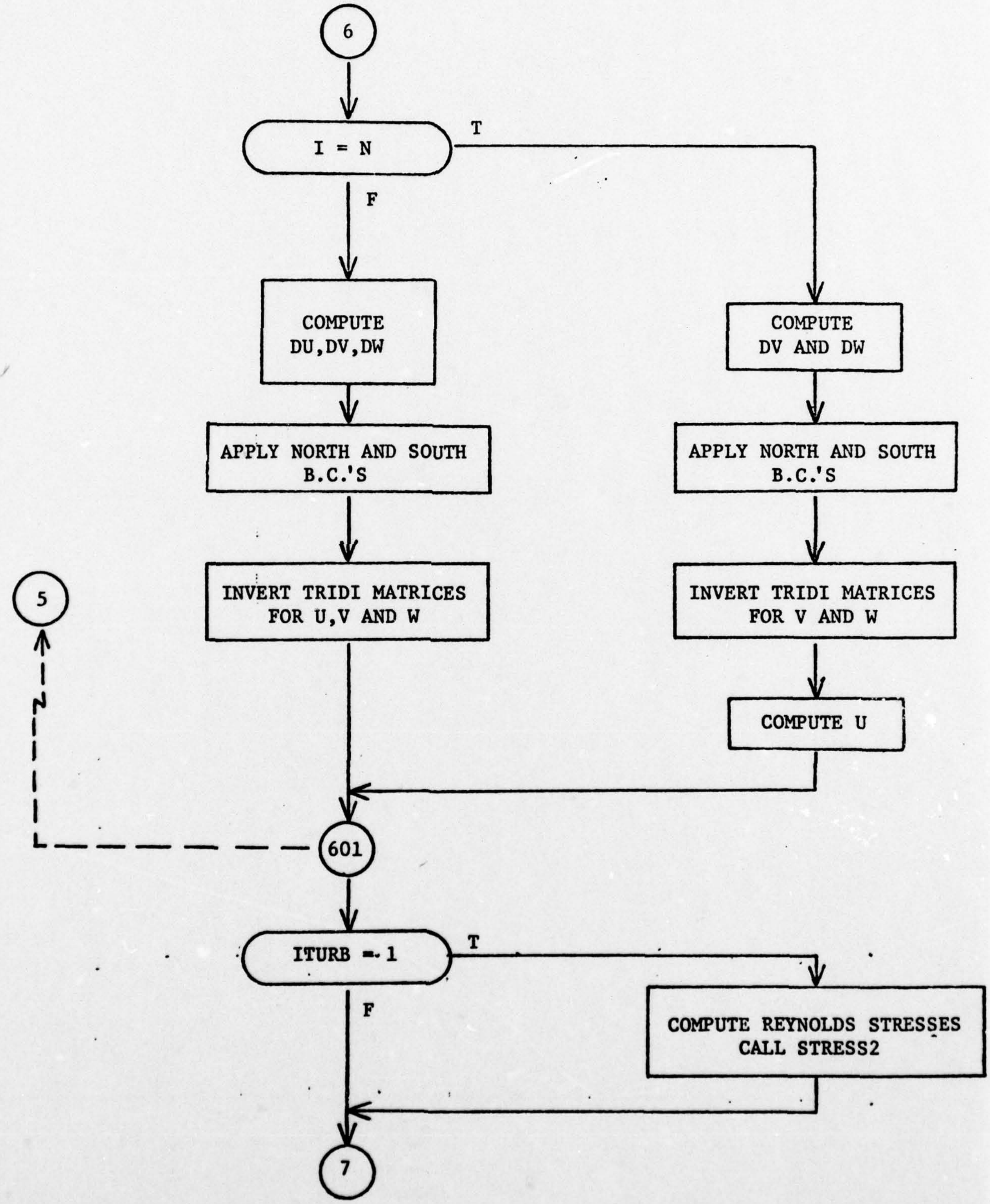
3.1 Main Program ICWAKE

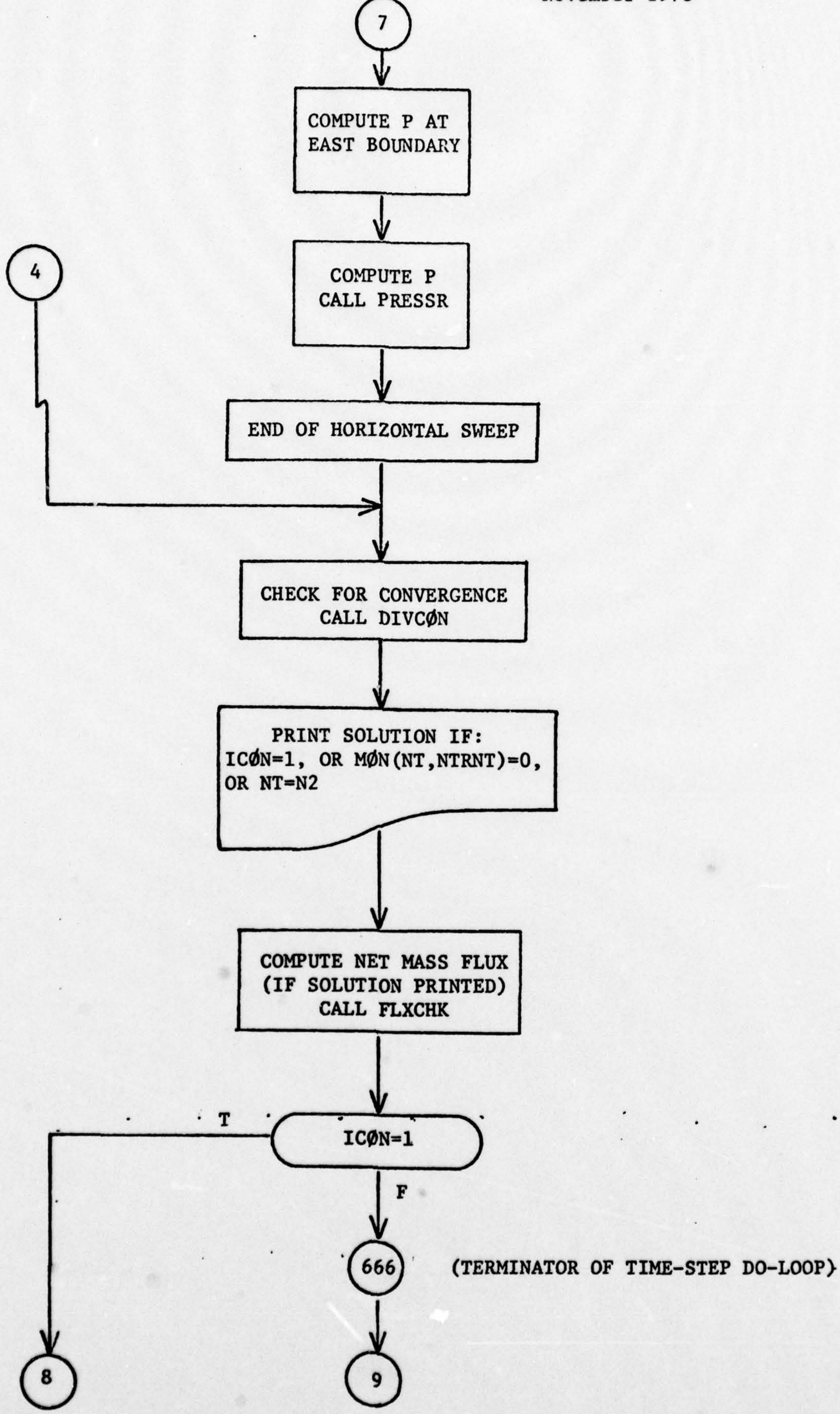
The main program ICWAKE controls the code. It handles nearly all input and output, and calls the subroutines that set up the finite-difference grid, evaluate the transformation coefficients, calculate the effect of a boundary layer and a propeller, and interpolate, from the input data, for the boundary values at the grid points along the WEST boundary. ICWAKE reads TAPE41 if data from a previous calculation is to be used as the initial conditions. ICWAKE performs the finite-difference solution of the mean-flow equations and computes the divergence of the velocity field; after each half time-step (i.e., after each ADI sweep), it calls subroutine STRESS2, which calculates Reynolds stresses, and subroutine PRESSR, which computes the pressure. ICWAKE calls routines that check for convergence, compute the net mass flux into the domain, and compare various terms in the equations of motion. At the end of a calculation, ICWAKE writes data on TAPE42, which may subsequently be used, through TAPE41, to restart the calculation or as initial conditions for a calculation.

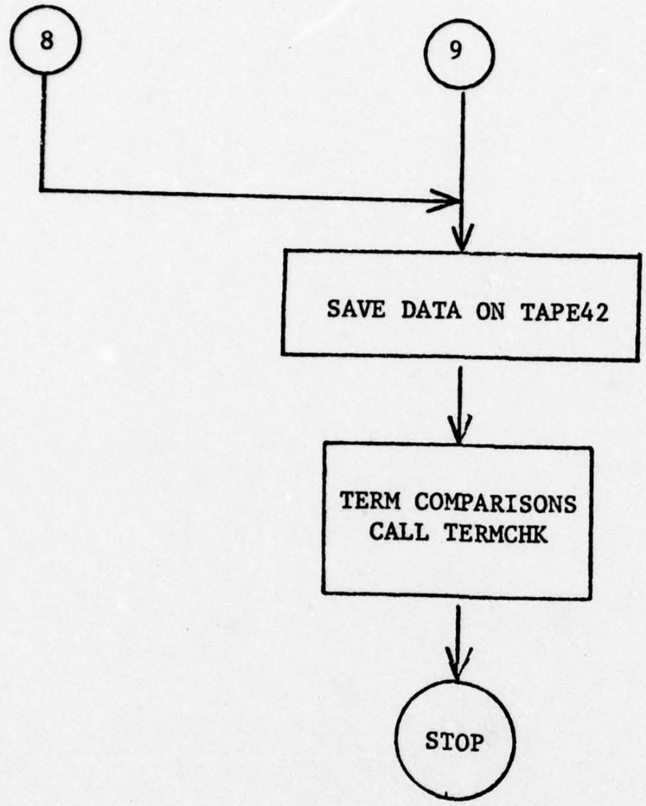

JWBL
READ $DAT1
NUMBER=1 NSTRT=0
F
T
READ TAPE41
READ $DAT2
T
NSTRT=0
F
READ $DAT3
T
IMAGER=1
F
READ $DAT4
T
IBLPRP=0
F
COMPUTE BOUNDARY LAYER AND PROPELLER INFLUENCE AT UPSTREAM BOUNDARY BY CALLING BLAYER, PRØPWV, PRØPU, PTURB
READ $DAT5 $DAT6
GENERATE GRID SYSTEM, COMPUTE TRANSFORMATION COEFFICIENTS, CALL CØEFF
COMPUTE INITIAL UPSTREAM CONDITIONS, CALL UPCØND
1


1
BEGIN ITERATIONS
DO 666 NT = N1,N2
T
ISWEEPY = 0
F
VERTICAL SWEEP
IMPLICIT IN X
DO 501J = 2,M-1
COMPUTE TRIDI COEFFICIENTS
AU,AV,AW,BU,...,CU,...DU...
2
APPLY EAST B.C.'S
INVERT MATRICES
FOR U,V,W
J = 2
T
F
SATISFY SOUTH B.C.
501
SATISFY NORTH B.C.
COMPUTE
DIV
3

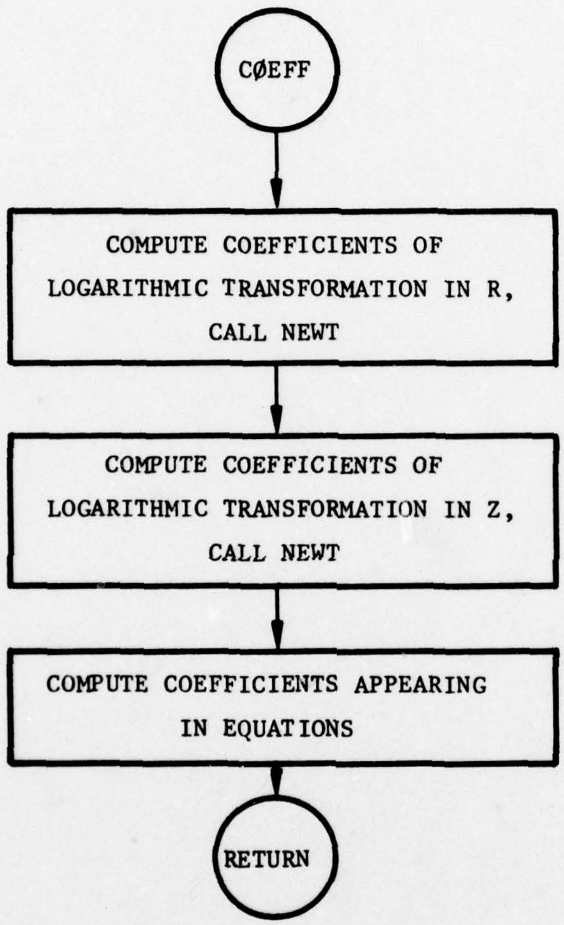

3
ITURB = 1
T
COMPUTE REYNOLDS STRESSES
CALL STRESS2
F
2
COMPUTE P AT
EAST BOUNDARY
COMPUTE P
CALL PRESSR
END OF VERTICAL SWEEP
T
ISWEEPX = 0
F
HORIZONTAL SWEEP
IMPLICIT TN Y
DØ 601 I = 2,N
4
COMPUTE TRIDI COEFFICIENTS
AU,...,BU,...,CU,...
5
6


6
I = N
T
F
COMPUTE
DU,DV,DW
COMPUTE
DV AND DW
APPLY NORTH AND SOUTH
B.C.'S
APPLY NORTH AND SOUTH
B.C.'S
5
INVERT TRIDI MATRICES
FOR U,V AND W
INVERT TRIDI MATRICES
FOR V AND W
COMPUTE U
601
ITURB = 1
T
F
COMPUTE REYNOLDS STRESSES
CALL STRESS2
7


7
COMPUTE P AT
EAST BOUNDARY
4
COMPUTE P
CALL PRESSR
END OF HORIZONTAL SWEEP
CHECK FOR CONVERGENCE
CALL DIVCØN
PRINT SOLUTION IF:
ICØN=1, OR MØN(NT,NTRNT)=0,
OR NT=N2
COMPUTE NET MASS FLUX
(IF SOLUTION PRINTED)
CALL FLXCHK
T
ICØN=1
F
666
(TERMINATOR OF TIME-STEP DO-LOOP)
8
9


8
9
SAVE DATA ON TAPE42
TERM COMPARISONS
CALL TERMCHK
STOP

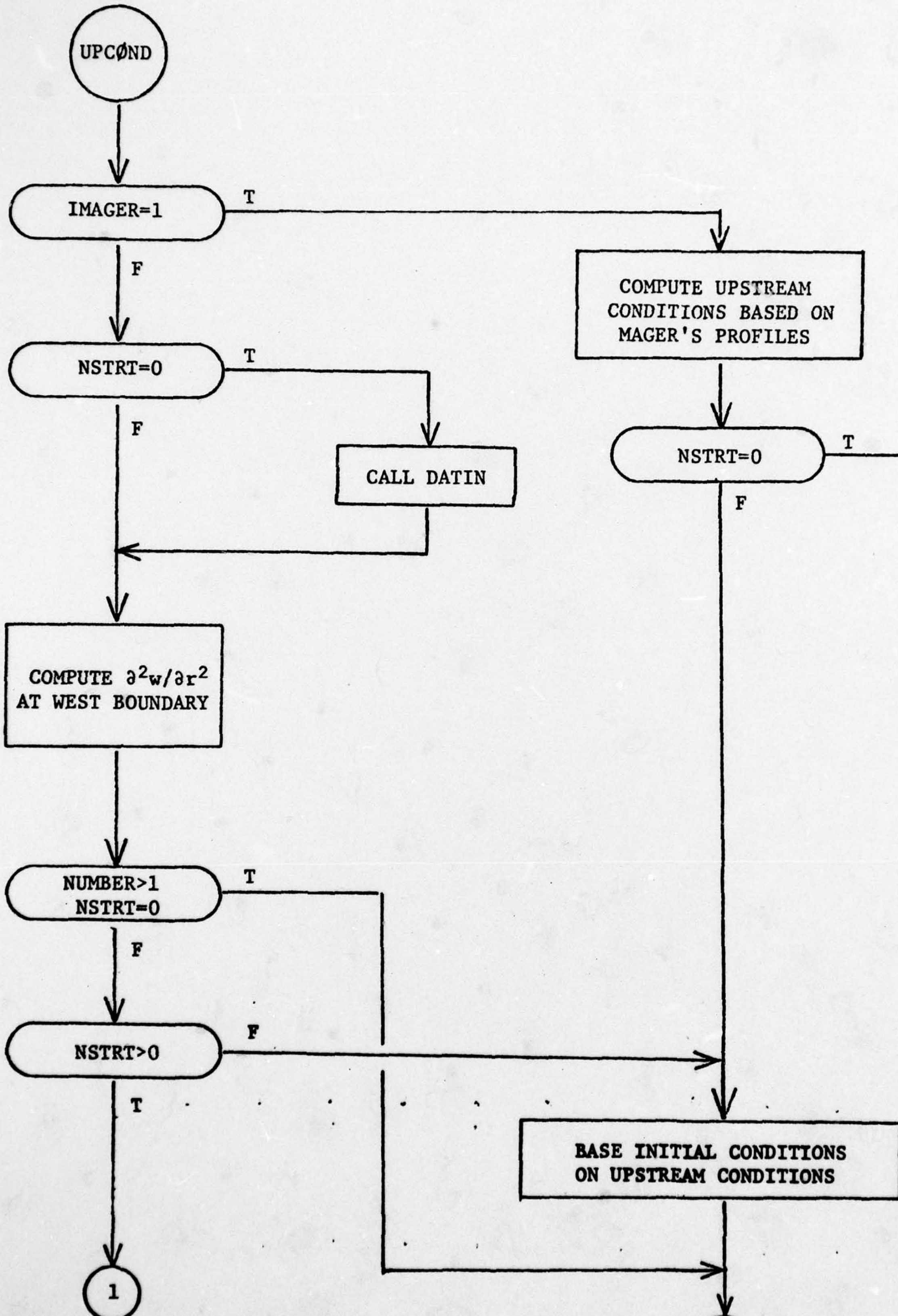
SUBROUTINE CØEFF. Subroutine CØEFF calculates the transformation coefficients that appear in the mean, turbulence model and in the pressure equations in transformed form. CØEFF calls subroutine NEWT, which computes the coefficients of the logarithmic transformations.

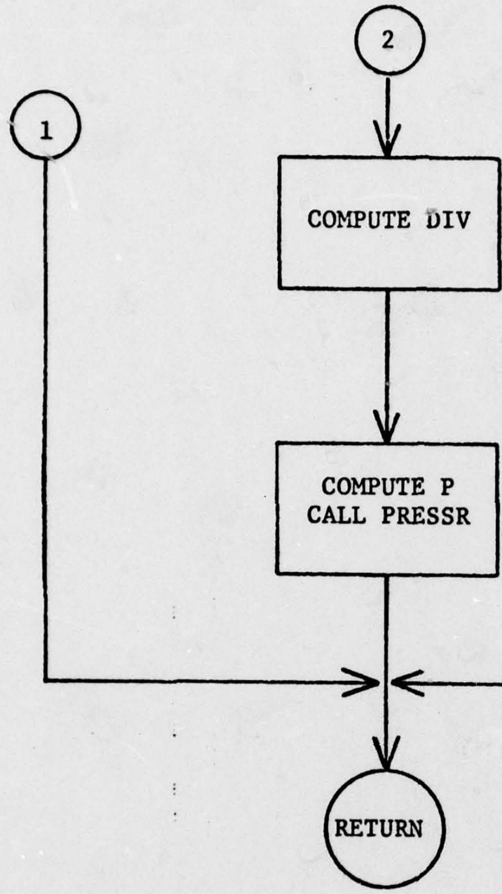


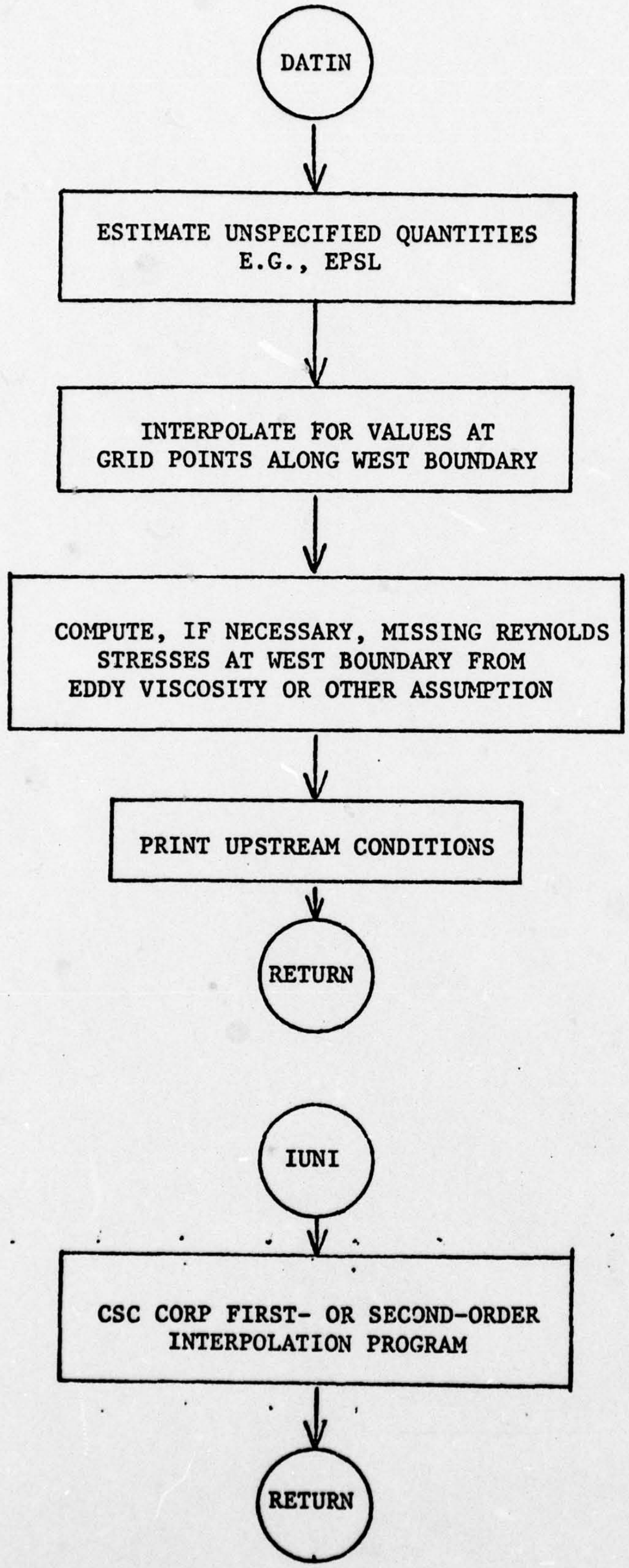
SUBROUTINE NEWT. Subroutine NEWT uses a Newton-Raphson technique to compute the coefficients of the logarithmic transformation based on the estimated values AX and AY, and on RCC, ZCC, ZINITL, ZMAX, RMAX, N, M, NC, and MC.

SUBROUTINES UPCØND, DATIN and IUNI. Subroutine UPCØND sets the initial conditions over the solution domain and computes $\partial^2 w/\partial r^2$ at the WEST boundary. If MAGER=1, Mager's cubic-quartic profiles are assumed upstream; otherwise, tabular data read into JWBL is assumed. If a solution from a previous calculation is not used as input to a new calculation (i.e., NUMBER=1, NSTRT=0), the conditions at the WEST boundary are applied everywhere, and subroutine PRESSER is called to provide the pressure field. If a calculation with new conditions (i.e., new Re or WEST boundary conditions) is begun with data from a previous calculation (i.e., NUMBER>1, NSTRT=0) as an initial condition, the solution at only the WEST boundary is changed. If a calculation is to be continued (i.e., NUMBER≥1, NSTRT>0) beginning with existing data from TAPE41, only $\partial^2 w/\partial r^2$ is computed. When NSTRT=0 and IMAGER=0 so that tabular data is assumed, subroutine DATIN is used to interpolate for the WEST boundary values at grid points.

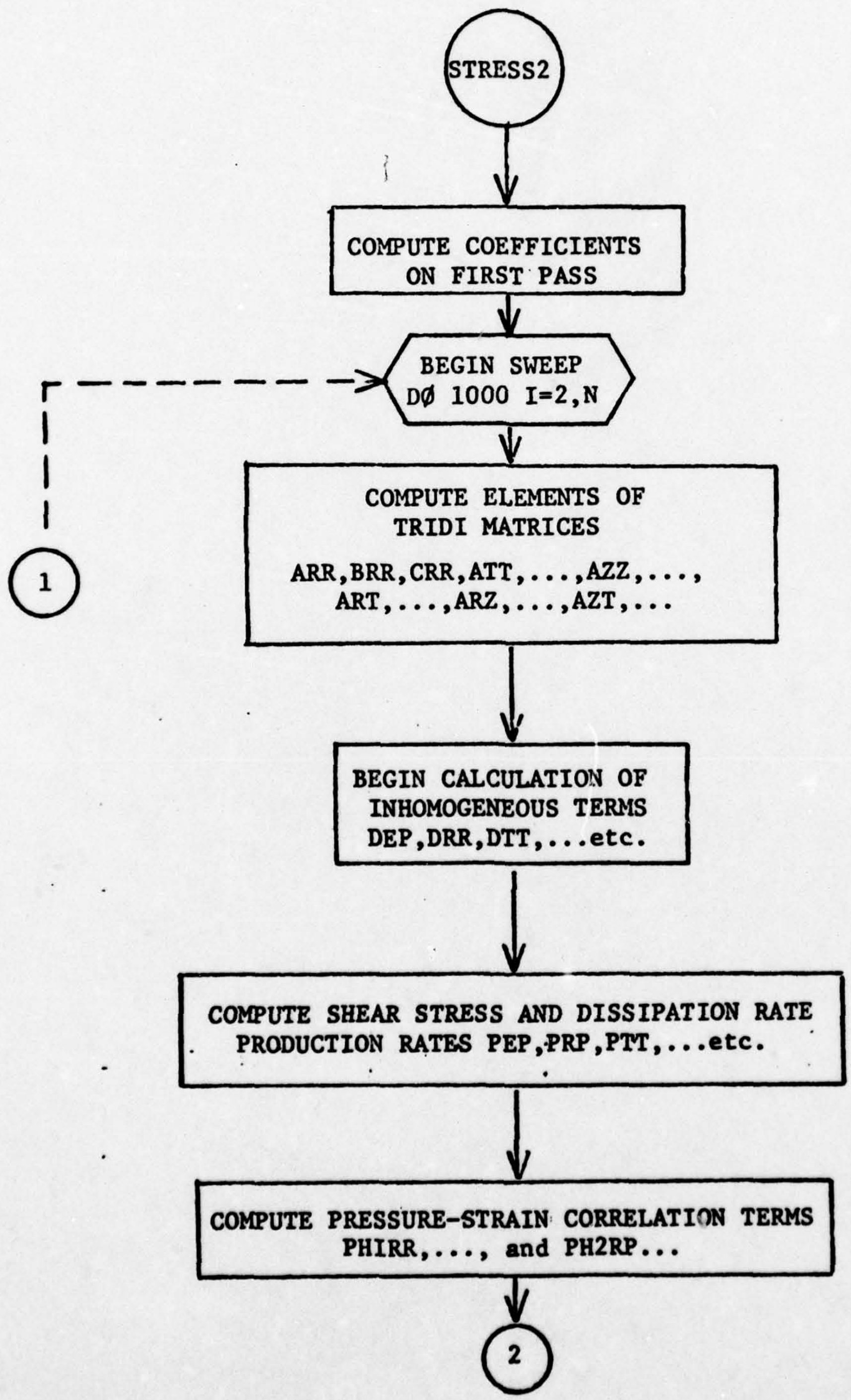
Subroutine DATIN calls IUNI, which performs either first- or second-order interpolation to calculate boundary values at the WEST grid points from tabular data. DATIN also estimates upstream conditions that are not specified, and prints out the upstream conditions.

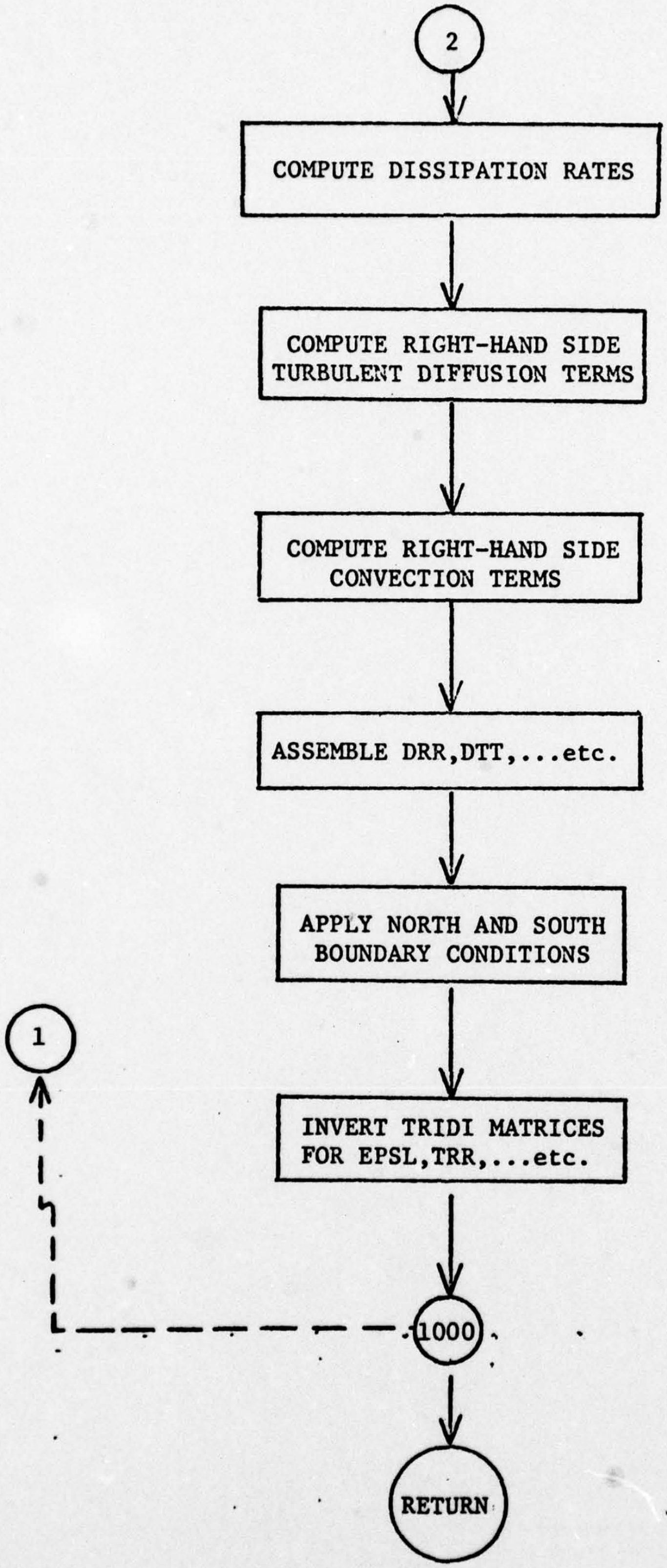

UPCØND
IMAGER=1
T
F
COMPUTE UPSTREAM CONDITIONS BASED ON MAGER'S PROFILES
NSTRT=0
T
F
CALL DATIN
NSTRT=0
T
F
3
COMPUTE ∂²w/∂r² AT WEST BOUNDARY
NUMBER>1
NSTRT=0
T
F
NSTRT>0
F
T
BASE INITIAL CONDITIONS ON UPSTREAM CONDITIONS
1
2


2
1
3
COMPUTE DIV
COMPUTE P
CALL PRESSR
RETURN

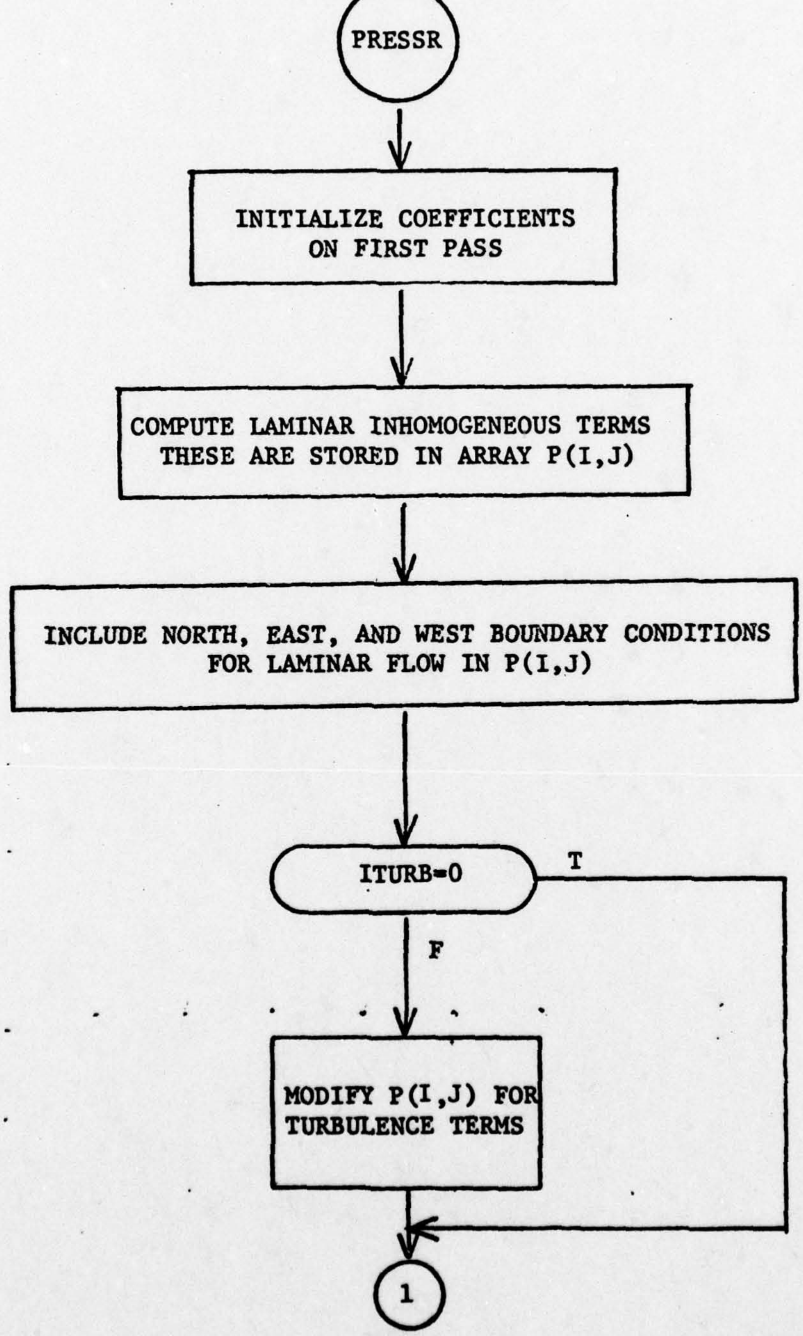

DATIN
ESTIMATE UNSPECIFIED QUANTITIES E.G., EPSL
INTERPOLATE FOR VALUES AT GRID POINTS ALONG WEST BOUNDARY
COMPUTE, IF NECESSARY, MISSING REYNOLDS STRESSES AT WEST BOUNDARY FROM EDDY VISCOSITY OR OTHER ASSUMPTION
PRINT UPSTREAM CONDITIONS
RETURN
IUNI
CSC CORP FIRST- OR SECOND-ORDER INTERPOLATION PROGRAM
RETURN

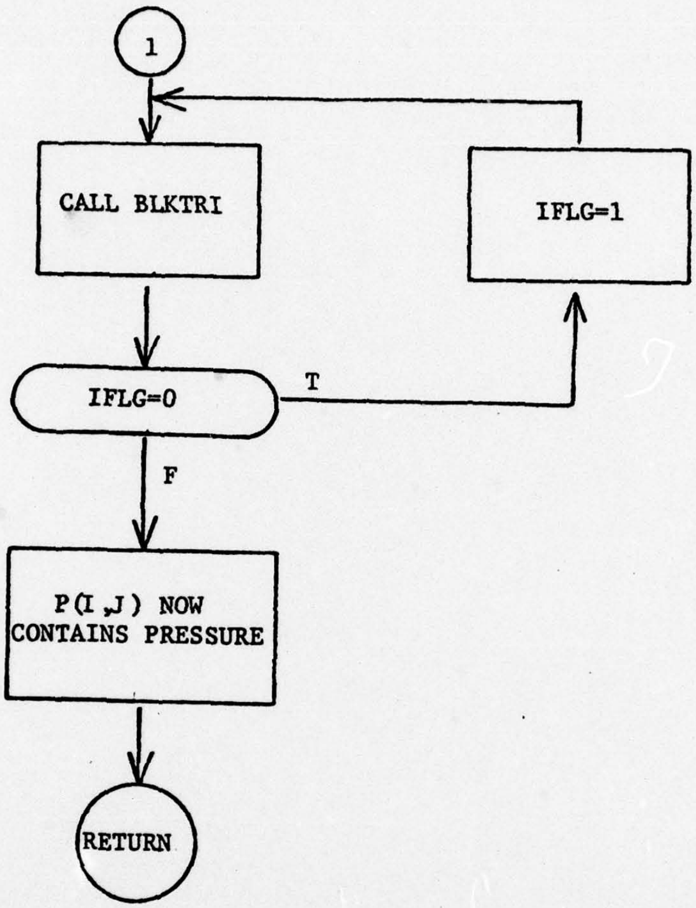
<u>SUBROUTINE STRESS2</u>. Subroutine STRESS2 computes Reynolds stresses using the second-order closure turbulence model of Hanjalic and Launder (1972) with the pressure/mean-strain correlation of Launder, Reece, and Rodi (1975). STRESS2 sweeps in x (implicit in y) only.




2
COMPUTE DISSIPATION RATES
COMPUTE RIGHT-HAND SIDE TURBULENT DIFFUSION TERMS
COMPUTE RIGHT-HAND SIDE CONVECTION TERMS
ASSEMBLE DRR,DTT,...etc.
APPLY NORTH AND SOUTH BOUNDARY CONDITIONS
1
INVERT TRIDI MATRICES FOR EPSL,TRR,...etc.
1000
RETURN

SUBROUTINE PRESSR. Subroutine PRESSR solves the Poisson equation for pressure obtained by taking the divergence of the momentum equations. It calls BLKTRI, the NCAR fast direct solver. A flow chart for BLKTRI is not presented here. Note that BLKTRI is initialized by calling in once with IFLG=0.




1
CALL BLKTRI
IFLG=1
IFLG=0
T
F
P(I,J) NOW CONTAINS PRESSURE
RETURN

SUBROUTINE TERMCHK. Subroutine TERMCHK is called at the end of each completed ICWAKE calculation. It computes and prints particular terms in the equations of motion, and the user can add terms as he wishes. TERMCHK allows the user to investigate the applicability of various approximations to the full equations in particular flow regions. At this time, TERMCHK calculates $\partial P/\partial z$, $\partial R_{zz}/\partial z$, $\partial R_{rz}/\partial r$, and R_{rz}/r at the axis. These terms appear in the axial (W) momentum equation. Because of the simplicity of this subroutine, a flow chart is not presented here.

SUBROUTINE NØZERØ. Subroutine NØZERØ is called by DATIN to ensure that the turbulence quantities R_{zz}, R_{rr}, $R_{\theta\theta}$, and ε are nonzero (although small) outside of the upstream wake boundary.

SUBROUTINE BLAYER. Subroutine BLAYER is called by ICWAKE to compute the profiles of the mean axial velocity and the seven turbulence quantities in the boundary layer aft of a body. Propeller effects are not included. See Grabowski, et al. (1976) for the details of the boundary layer algorithm.

SUBROUTINE PRØPWV. Subroutine PRØPWV is called by ICWAKE to compute the profiles of the mean axial and swirl velocities immediately behind a propeller that is attached to a body. The mean axial velocity profile computed by subroutine BLAYER is assumed to represent the flow entering the propeller. See Schwartz and Bernstein (1975) for the details of the propeller algorithm.

SUBROUTINE PRØPU. Subroutine PRØPU is called by ICWAKE to compute the radial velocity profile immediately behind a propeller that is attached to a body. The calculation is based on the mean axial velocity profile compated by PRØPWV. See Grabowski, et al. (1976) for the details of the radial velocity calculation.

SUBROUTINE PTURB. Subroutine PTURB is called by ICWAKE to compute the modification of the turbulence in the boundary layer aft of a body as it passes through a propeller attached to the body. The turbulence profiles computed by BLAYER are assumed to represent the turbulence coming into the propeller. See Grabowski, et al. (1976) for the details of the turbulence modification algorithm.

4. USE OF ICWAKE - General Observations

As we pointed out in the Introduction, ICWAKE obtains a finite-difference solution for a rather complicated, lengthy, elliptic system of eleven, coupled, partial differential equations. Although this manual should make application of the code as easy as possible, the inexperienced user should be prepared for some initial difficulty. Several observations may prove useful.

First, we note that no formal stability analysis has been applied to the finite-difference equations, and no theoretical criterion for either the maximum or optimum time-step (if such exists) is available. However, since the finite-difference representation of the equations for the mean flow and turbulence quantities is based on generally stable implicit formulations, we expect, and find empirically, reasonably good stability properties. In general, the maximum time-step that may be stably applied is at least two to three times the smallest mesh width in r-z space, and it is relatively insensitive to the free-stream Reynolds number when upwind differencing is applied to the axial convection terms. A good procedure to follow in selecting a time-step is to set it to the smallest r-z space mesh width and, in a sequence of calculations, which need not be performed to convergence, gradually increase it. Instability caused by too large a time-step is usually catastrophic and readily apparent; however, we have encountered, in laminar flow calculations, slowly growing "wiggles" in the calculations for the upstream portion of the domain. These "wiggles" were smoothed in subsequent calculations with a smaller time-step.

The second observation is that, at sufficiently high Reynolds numbers when centered differences are used for the axial convection terms, growing or stationary "wiggles" will often appear in the downstream portion of the domain in calculations. Roache (1972) discusses this behavior. Increasing the mesh resolution will usually rectify this problem, although in most cases the only practical solution is to switch to upwind differencing.

A third point is that both careful consideration and experimentation are required in the arrangement of the computational domain and finite-difference grid. We require that RMAX and ZMAX, which locate the NORTH and EAST boundaries, be chosen sufficiently large that

choosing them larger would have no substantial effect on the computed solution in the interior of the domain. At the same time we would like the domain to be as small as possible in order to maximize the possible resolution with a given, limited number of grid points. Thus, for a given flow or a given class of flows, some iteration is necessary to arrive at the optimum value of RMAX and ZMAX.

For "production" calculations, we recommend a grid of 60 x 32 grid points (N = 60, M = 32). For debug and trend calculations, N can be set to 40 and M to 16. Optimum choice of the grid parameters RCC, ZCC, NC, and MC depends very much on the nature of the flow to be computed, so some physically based insight may be useful. For many calculations, the following values are effective: RCC = 1, ZCC = ZINITL + 1, MC = 8 or 14 when M = 16 or 32, respectively, and NC = 10 or 14 when N = 40 or 60, respectively. These values assume that RMAX is between about 4 and 8 and that ZMAX is between ZINITL+10 and ZINITL+20. Flows with high swirl and large axial gradients in the upstream portion of the domain may require larger values of NC.

5. SAMPLE CALCULATION

The sample calculation in this section in based on the combined data of Swanson, et al. (1974) and Chieng, et al. (1974). These data were obtained in the wake behind a blunt-nosed, parallel-sided, sharp-sterned body with a ratio between the length and maximum diameter of 6. The flow Reynolds number, based on the free-stream velocity and body radius, was 3.1×10^5. The body was fitted with a 6-inch-diameter aft-mounted propeller, and the experiments were conducted under approximately drag-free conditions. Hot-wire data were obtained at axial positions 4, 10, 20, 40, and 80 body radii downstream of the body. These include R_{rr}, $R_{\theta\theta}$, R_{zz}, R_{rz}, and $R_{\theta z}$*, as well as the axial and circumferential mean velocity components, W and V.

In the test case included here, we have used the experimental data at 4 body radii behind the body to compute the flow in the region behind the body to a location 20 body radii downstream. The radial boundary of the computational domain was set at 4 body radii. Upstream conditions for three flow variables were not measured in the experiment and were therefore estimated for this calculation as follows: The radial mean velocity U was set to zero; the $R_{r\theta}$ turbulence correlation was estimated from an eddy viscosity that was obtained from the known or calculable values of R_{rz} and $\partial W/\partial r$; the dissipation rate ε was assumed to equal $K_\varepsilon k^{3/2}/\ell$, where $K = \frac{1}{2}(R_{rr} + R_{\theta\theta} + R_{zz})$, $K_\varepsilon + 0.53$, and $\ell + 0.2$ (i.e., 0.2 body radii). The values of the constant K_ε and the turbulence integral scale ℓ were suggested by Gran (1976).

We wish to emphasize that the computation presented here should be considered as a sample calculation, not as an accurate prediction of a real flow. Considerable testing and evaluation will be necessary before we will be able to place confidence in ICWAKE predictions.

THE COMPUTER CODE OUTPUT INCLUDED IN THIS SECTION HAS BEEN ABRIDGED.

---

*This notation is defined in Grabowski (1976).

Test Case Input Data

```
#
$DAT1
 M11=1,M22=20,M33=1,
 TA=0.1,
 NTMX=400,
 NPRNT=100,
 NTCHK=5,
$END
$DAT2
 N=60,M=32,NC=12,MC=18,
 ZINITL=4.,ZCC=5.,
 ZMAX=20.,
$END
$DAT3
 ITURB=1,
 RE=310000.,
$END
$DAT4
 MPOINT=22,
 RT(1)=0.,.083,0.167,0.250,0.333,0.417,0.5,0.583,0.667,0.750,0.833,0.917,
  1.,1.083,1.167,1.250,1.333,1.50,1.667,2.0,2.667, 4.,
 UU(1)=3.50E-3,3.52E-3,4.20E-3,5.91E-3,5.76E-3,4.29E-3,3.01E-3,2.17E-3,
  1.43E-3,1.66E-3,3.28E-3,1.05E-2,4.53E-3,1.32E-3,5.20E-4,2.37E-4,1.32E-4,
  5.48E-5,4.90E-5,3.84E-5,2.70E-5,0.,
 VV(1)=3.50E-3,3.52E-3,4.20E-3,5.61E-3,6.46E-3,5.55E-3,5.21E-3,3.61E-3,
  1.89E-3,1.28E-3,1.40E-3,4.54E-3,1.76E-3,4.84E-4,1.90E-4,1.02E-4,5.63E-5,
  3.36E-5,2.81E-5,2.60E-5,2.60E-5,0.,
 WW(1)=2.55E-3,2.62E-3,3.45E-3,4.28E-3,4.58E-3,3.84E-3,2.92E-3,1.94E-3,
  1.12E-3,1.08E-3,1.63E-3,3.03E-3,2.00E-3,7.40E-4,2.43E-4,1.12E-4,
 5.33E-5,1.44E-5,1.09E-5,9.61E-6,0.,0.,
 WU(1)=0.00E-4,-8.52E-4,-1.52E-3,-2.31E-3,-2.43E-3,-1.92E-3,-1.33E-3,-6.92E-4,
 -1.13E-4,2.73E-4,3.96E-4,9.2E-5,1.16E-4,-3.E-5,2.2E-5,5.2E-6,3.9E-6,
  2.4E-6,5.E-7,3.E-7,0.,0.,
 KV(1)=0.E-4,2.0E-4,5.1E-4,8.5E-4,1.88E-3,2.E-3,1.8E-3,8.1E-4,8.E-6,
  -4.7E-4,-1.13E-3,-1.38E-3,-1.58E-3,-4.4E-4,-1.77E-4,-8.E-5,-3.1E-5,
 -5.2E-6,-3.3E-6,-1.8E-6,-9.E-7,0.,
 WL(1)=.688,.701,.733,.796,.860,.924,.989,1.045,1.075,1.089,1.079,1.045,1.006,
 1.003,1.0,1.0,1.0, 5*1.0,
 VL(1)=.0000,.0139,.0330,.0551,.0722,.0789,.0856,.0765,.0680,.0543,.0373,
  .0142,-.022,-.052,-.0052,-.0052,-.005,5*0.,
 UL(1)=30*0.,
$END
$DAT5
$END
$DAT6
 ARTVIS(1)=60*1.,
 $END
$DAT8
 NML=1,
$END
```

```
A R P A   I C W A K E   P R O G R A M - J W B L   C O D E

W. J.  G R A B O W S K I          06 JUN 76       10.21.48


NUMERICAL SOLUTION OF INCOMPRESSIBLE,AXISYMMETRIC NAVIER-STOKES EQUATIONS
FOR SWIRLING FLOWS WITH LARGE AXIAL GRADIENTS

CENTERED-UPWIND DIFFERENCING USING VARIABLE ARTIFICAL VISCOSITY
DIRECT SOLUTION FOR PRESSURE
PARABOLIC OUTFLOW BOUNDARY CONDITION
```

```
SEQUENCE NUMBER =     1

ITERATION COUNT AT START, NSTRT =      0

MAX NUMBER OF ITERATIONS IN THIS RUN, NTMX =  400       PRINT INTERVAL, NPRNT =   100

RADIAL PRINT PARAMETER     M1 =    1     M2 =    1     M3 =   1

GRID POINTS IN X, N =    60      GRID POINTS IN Y, M =    32

GRID SIZE IN X, H =  .169492E-01      GRID SIZE IN Y, K =  .161290E-01

ZMAX =  .2000E+02     RMAX =  .4000E+01

XMAX =  .100000E+01          YMAX =  .500000E+00


COORDINATE TRANSFORMATION PARAMETERS
NC =   12      MC =   18      ZCC =  .5000E+01      RCC =  .1000E+01      ZINITL =  .4000E+01
AX =  .2219E+01      BX =  .1951E+01      AY =  .5280E+01      BY =  .3073E+00      EPS =  .1000E-05
```

Z TO X TRANSFORMATION

I	X	Z	XH	ZH
1	0.	.40000E+01	.84746E-02	.40370E+01
2	.16949E-01	.40748E+01	.25424E-01	.41133E+01
3	.33898E-01	.41525E+01	.42373E-01	.41924E+01
4	.50847E-01	.42331E+01	.59322E-01	.42746E+01
5	.67797E-01	.43168E+01	.76271E-01	.43599E+01
6	.84746E-01	.44038E+01	.93220E-01	.44485E+01
7	.10169E+00	.44940E+01	.11017E+00	.45404E+01
8	.11864E+00	.45878E+01	.12712E+00	.46360E+01
9	.13559E+00	.46851E+01	.14407E+00	.47351E+01
10	.15254E+00	.47861E+01	.16102E+00	.48381E+01
11	.16949E+00	.48911E+01	.17797E+00	.49450E+01
12	.18644E+00	.50000E+01	.19492E+00	.50560E+01
13	.20339E+00	.51131E+01	.21186E+00	.51713E+01
14	.22034E+00	.52306E+01	.22881E+00	.52910E+01
15	.23729E+00	.53525E+01	.24576E+00	.54153E+01
16	.25424E+00	.54792E+01	.26271E+00	.55443E+01
17	.27119E+00	.56107E+01	.27966E+00	.56783E+01
18	.28814E+00	.57472E+01	.29661E+00	.58174E+01
19	.30508E+00	.58890E+01	.31356E+00	.59619E+01
20	.32203E+00	.60361E+01	.33051E+00	.61118E+01
21	.33898E+00	.61890E+01	.34746E+00	.62676E+01
22	.35593E+00	.63477E+01	.36441E+00	.64293E+01
23	.37288E+00	.65125E+01	.38136E+00	.65972E+01
24	.38983E+00	.66836E+01	.39831E+00	.67715E+01
25	.40678E+00	.68612E+01	.41525E+00	.69526E+01
26	.42373E+00	.70457E+01	.43220E+00	.71405E+01
27	.44068E+00	.72372E+01	.44915E+00	.73357E+01
28	.45763E+00	.74361E+01	.46610E+00	.75383E+01
29	.47458E+00	.76426E+01	.48305E+00	.77488E+01
30	.49153E+00	.78570E+01	.50000E+00	.79672E+01
31	.50847E+00	.80796E+01	.51695E+00	.81941E+01
32	.52542E+00	.83107E+01	.53390E+00	.84296E+01
33	.54237E+00	.85508E+01	.55085E+00	.86742E+01
34	.55932E+00	.88000E+01	.56780E+00	.89282E+01
35	.57627E+00	.90588E+01	.58475E+00	.91918E+01
36	.59322E+00	.93274E+01	.60169E+00	.94656E+01
37	.61017E+00	.96064E+01	.61864E+00	.97499E+01
38	.62712E+00	.98961E+01	.63559E+00	.10045E+02
39	.64407E+00	.10197E+02	.65254E+00	.10352E+02
40	[illegible]	.10509E+02	.66949E+00	.10670E+02
41	[illegible]	[illegible]	.68644E+00	.11000E+02
42	[illegible]	[illegible]	.70339E+00	.11343E+02
43	[illegible]	[illegible]	.72034E+00	.11700E+02
44	[illegible]	[illegible]	.73729E+00	.12070E+02
45	[illegible]	[illegible]	[illegible]	.12454E+02
46	[illegible]	[illegible]	[illegible]	.12852E+02
47	[illegible]	[illegible]	[illegible]	.13267E+02
48	[illegible]	[illegible]	[illegible]	.13697E+02
49	[illegible]	[illegible]	[illegible]	.14143E+02
50	[illegible]	[illegible]	[illegible]	.14607E+02
51	[illegible]	[illegible]	[illegible]	.15088E+02
52	[illegible]	[illegible]	[illegible]	.15588E+02
53	[illegible]	[illegible]	[illegible]	.16107E+02
54	[illegible]	[illegible]	[illegible]	.16645E+02
55	[illegible]	[illegible]	.92373E+00	.17205E+02
56	[illegible]	[illegible]	[illegible]	.17746E+02
57	[illegible]	[illegible]	.95763E+00	.18380E+02
58	[illegible]	.18699E+02	.97458E+00	.19015E+02
59	[illegible]	[illegible]	[illegible]	.19666E+02

58	.96610E+00	[illegible]	[illegible]	[illegible]
59	.98305E+00	.19337E+02	.99153E+00	.19666E+02
60	.10000E+01	.20000E+02	IIIII	IIIII

R TO Y TRANSFORMATUON

J	Y	R	RW
1	0.	0.	.13370E-01
2	.16129E-01	.27321E-01	.41879E-01
3	.32258E-01	.57070E-01	.72922E-01
4	.48387E-01	.89464E-01	.10673E+00
5	.64516E-01	.12474E+00	.14353E+00
6	.80645E-01	.16315E+00	.18361E+00
7	.96774E-01	.20497E+00	.22726E+00
8	.11290E+00	.25051E+00	.27478E+00
9	.12903E+00	.30011E+00	.32653E+00
10	.14516E+00	.35411E+00	.38288E+00
11	.16129E+00	.41291E+00	.44424E+00
12	.17742E+00	.47693E+00	.51105E+00
13	.19355E+00	.54665E+00	.58380E+00
14	.20968E+00	.62257E+00	.66302E+00
15	.22581E+00	.70524E+00	.74929E+00
16	.24194E+00	.79525E+00	.84322E+00
17	.25806E+00	.89327E+00	.94550E+00
18	.27419E+00	.10000E+01	.10569E+01
19	.29032E+00	.11162E+01	.11781E+01
20	.30645E+00	.12428E+01	.13102E+01
21	.32258E+00	.13806E+01	.14540E+01
22	.33871E+00	.15306E+01	.16106E+01
23	.35484E+00	.16940E+01	.17811E+01
24	.37097E+00	.18719E+01	.19667E+01
25	.38710E+00	.20657E+01	.21689E+01
26	.40323E+00	.22766E+01	.23890E+01
27	.41935E+00	.25063E+01	.26287E+01
28	.43548E+00	.27565E+01	.28897E+01
29	.45161E+00	.30288E+01	.31740E+01
30	.46774E+00	.33254E+01	.34834E+01
31	.48387E+00	.36483E+01	.38204E+01
32	.50000E+00	.40000E+01	IIIII

```
 $DAT3
 RE      =  0.31E+06,
 WI      = 1.,
 VI      = 1.,
 ALPH    = 1.,
 IMAGER  =  0,
 ITURB   =  1,
 IDECOUP =  0,
 $END
```

```
 SDATS
  ISWEEPX =  1,
  ISWEEPY =  1,
 SEND
```

```
$DAT6
ARTVIS  =  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
           0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
           0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
           0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
           0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
          1.
$END
```

```
THE FLOW REYNOLDS NUMBER IS BASED ON A CHARACTERISTIC RADIUS, E.G. BODY OR NOZZLE RADIUS
AND A CHARACTERISTIC AXIAL MEAN VELOCITY, E.G. THE FREE-STREAM VELOCITY

IN THIS CALCULATION RE =  .3100E+06


THE EQUATION SYSTEM WILL BE MARCHED WITH TIME STEP TA =  .1000E+00



R AND Z ARE RADIAL AND AXIAL COORDINATES
NON-DIMENSIONALIZED BY THE CHARACTERISTIC LENGTH

U, V, AND W ARE VELOCITY IN THE RADIAL, CIRCUMFERENTIAL AND AXIAL DIRECTIONS
NON-DIMENSIONALIZED BY THE CHARACTERISTIC VELOCITY

P IS PRESSURE NORMALIZED BY ITS VALUE AT POINT  N,M

DIV IS THE DIVERGENCE OF THE VELOCITY FIELD


IN CALCULATIONS WITH TURBULENCE, T-TERMS SUCH AS TRR,TTT,TZZ REPRESENT
TURBULENT CORRELATIONS I.E. NEGATIVE REYNOLDS STRESSES
```

```
 $DAT7
 MPOINT  =  22,
 RT      =  0.0,  0.83E-01,  0.167E+00,  0.25E+00,  0.333E+00,  0.417E+00,        0.5E+00,  0.583E+00,  0.667E+00,  0.75E+00,
            0.833E+00,  0.917E+00,  0.1E+01,  0.1083E+01,  0.1167E+01,          0.125E+01,  0.1333E+01,  0.15E+01,  0.1667E+01,
            0.2E+01,  0.2667E+01,  0.4E+01, I, I, I, I, I, I, I, I, I,
 UL      =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,     0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
            0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0, I,   I,
 VL      =  0.0,  0.139E-01,  0.33E-01,  0.551E-01,  0.722E-01,  0.789E-01,      0.856E-01,  0.765E-01,  0.68E-01,  0.543E-01,
            0.373E-01,  0.142E-01, -0.22E-01, -0.52E-01, -0.52E-02, -0.52E-02,  -0.5E-02,  0.0,  0.0,  0.0,  0.0,  0.0, I, I, I,
           I, I, I, I, I, I, I,
 WL      =  0.688E+00,  0.701E+00,  0.733E+00,  0.796E+00,  0.86E+00,          0.924E+00,  0.989E+00,  0.1045E+01,  0.1075E+01,
            0.1089E+01,  0.1079E+01,  0.1045E+01,  0.1006E+01,  0.1003E+01,     0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
            0.1E+01,  0.1E+01,  0.1E+01, I, I, I, I, I, I, I, I, I,
 UU      =  0.35E-02,  0.352E-02,  0.42E-02,  0.591E-02,  0.576E-02,  0.429E-02,  0.301E-02,  0.217E-02,  0.143E-02,  0.166E-02,
            0.328E-02,  0.105E-01,  0.453E-02,  0.132E-02,  0.52E-03,           0.237E-03,  0.132E-03,  0.548E-04,  0.49E-04,
            0.384E-04,  0.27E-04,  0.0, I, I, I, I, I, I, I, I, I,
 VV      =  0.35E-02,  0.352E-02,  0.42E-02,  0.561E-02,  0.646E-02,  0.555E-02,  0.521E-02,  0.361E-02,  0.189E-02,  0.128E-02,
            0.14E-02,  0.454E-02,  0.176E-02,  0.484E-03,  0.19E-03,  0.102E-03,   0.563E-04,  0.336E-04,  0.281E-04,  0.26E-04,
            0.26E-04,  0.0, I, I, I, I, I, I, I, I, I,
 WW      =  0.255E-02,  0.262E-02,  0.345E-02,  0.428E-02,  0.458E-02,          0.384E-02,  0.292E-02,  0.194E-02,  0.112E-02,
            0.108E-02,  0.163E-02,  0.303E-02,  0.2E-02,  0.74E-03,  0.243E-03,   0.112E-03,  0.533E-04,  0.144E-04,  0.109E-04,
            0.961E-05,  0.0,  0.0, I, I, I, I, I, I, I, I, I,
 WU      =  0.0, -0.852E-03, -0.152E-02, -0.231E-02, -0.243E-02, -0.192E-02,    -0.133E-02, -0.692E-03, -0.113E-03,  0.273E-03,
            0.396E-03,  0.92E-04,  0.116E-03, -0.3E-04,  0.22E-04,  0.52E-05,    0.39E-05,  0.24E-05,  0.5E-06,  0.3E-06,  0.0,
            0.0, I, I, I, I, I, I, I, I, I,
 WV      =  0.0,  0.2E-03,  0.51E-03,  0.85E-03,  0.188E-02,  0.2E-02,          0.18E-02,  0.81E-03,  0.8E-05, -0.47E-03,
           -0.113E-02, -0.138E-02, -0.158E-02, -0.44E-03, -0.177E-03, -0.8E-04,  -0.31E-04, -0.52E-05, -0.33E-05, -0.18E-05,
           -0.9E-06,  0.0, I, I, I, I, I, I, I, I, I,
 UV      =  0.0, -0.326E-03, -0.505E-03, -0.73E-03, -0.275E-03,                 0.39999999999998E-04,  0.235E-03,  0.59E-04,
           -0.525E-04, -0.985E-04, -0.367E-03, -0.644E-03, -0.732E-03,          -0.235E-03, -0.775E-04, -0.374E-04, -0.1355E-04,
           -0.14E-05, -0.14E-05, -0.75E-06, -0.45E-06,  0.0, I, I, I, I, I, I,   I, I, I, I,
 EPSIL   =  0.87439152144416E-03,  0.88954222901866E-03,  0.12085883565167E-02,  0.18607435012655E-02,  0.20401626993943E-02,
            0.14990980060823E-02,  0.11016120218083E-02,  0.63551650038374E-03,  0.27718858125832E-03,  0.23880309990136E-03,
            0.46961819968426E-03,  0.22758210458595E-02,  0.70718440715209E-03,  0.12021994941805E-03,  0.27563872438588E-04,
            0.89735790563707E-05,  0.35184086146325E-05,  0.97654115875368E-06,  0.77343690111087E-06,  0.59653576508522E-06,
            0.36150531332254E-06,  0.0, I, I, I, I, I, I, I, I, I,
 $END
```

UPSTREAM CONDITIONS J, R, U, V, W

1	0.	0.	0.	.69073E+00
2	.27321E-01	0.	.45754E-02	.69228E+00
3	.57070E-01	0.	.95575E-02	.69694E+00
4	.89464E-01	0.	.15370E-01	.70346E+00
5	.12474E+00	0.	.23390E-01	.71690E+00
6	.16315E+00	0.	.32124E-01	.73153E+00
7	.20497E+00	0.	.43111E-01	.76182E+00
8	.25051E+00	0.	.55206E-01	.79640E+00
9	.30011E+00	0.	.65423E-01	.83464E+00
10	.35411E+00	0.	.73883E-01	.87608E+00
11	.41291E+00	0.	.78573E-01	.92088E+00
12	.47693E+00	0.	.83738E-01	.97094E+00
13	.54665E+00	0.	.80485E-01	.10205E+01
14	.62257E+00	0.	.72496E-01	.10591E+01
15	.70524E+00	0.	.61689E-01	.10814E+01
16	.79525E+00	0.	.45032E-01	.10835E+01
17	.89327E+00	0.	.20726E-01	.10546E+01
18	.10000E+01	0.	-.22000E-01	.10060E+01
19	.11162E+01	0.	-.33492E-01	.10018E+01
20	.12428E+01	0.	-.52000E-02	.10000E+01
21	.13806E+01	0.	-.35757E-02	.10000E+01
22	.15306E+01	0.	0.	.10000E+01
23	.16940E+01	0.	0.	.10000E+01
24	.18719E+01	0.	0.	.10000E+01
25	.20657E+01	0.	0.	.10000E+01
26	.22766E+01	0.	0.	.10000E+01
27	.25063E+01	0.	0.	.10000E+01
28	.27565E+01	0.	0.	.10000E+01
29	.30288E+01	0.	0.	.10000E+01
30	.33254E+01	0.	0.	.10000E+01
31	.36483E+01	0.	0.	.10000E+01
32	.40000E+01	0.	0.	.10000E+01

UPSTREAM CONDITIONS ... J, TRR, TTT, TZZ, TRT, TRZ, TZT, EPSL

1	.35042E-02	.35042E-02	.25647E-02	0.	0.	0.	.87757E-03
2	.35066E-02	.35066E-02	.25730E-02	-.52112E-06	-.28045E-03	.65833E-04	.87938E-03
3	.35138E-02	.35138E-02	.25981E-02	-.20946E-04	-.58583E-03	.13752E-03	.88481E-03
4	.35723E-02	.35723E-02	.26839E-02	-.10055E-03	-.90341E-03	.22386E-03	.91409E-03
5	.38579E-02	.38579E-02	.30324E-02	-.12459E-03	-.11639E-02	.35403E-03	.10481E-02
6	.41688E-02	.41688E-02	.34119E-02	-.13066E-03	-.14894E-02	.49578E-03	.11940E-02
7	.49823E-02	.48451E-02	.38297E-02	-.13716E-03	-.18814E-02	.66555E-03	.15069E-02
8	.59091E-02	.56153E-02	.42819E-02	-.43573E-04	-.23107E-02	.85639E-03	.18619E-02
9	.58194E-02	.61231E-02	.44611E-02	.11595E-03	-.23824E-02	.14718E-02	.19691E-02
10	.53907E-02	.62314E-02	.43941E-02	.27642E-03	-.23019E-02	.19102E-02	.19042E-02
11	.43616E-02	.55944E-02	.38761E-02	.27660E-03	-.19449E-02	.19942E-02	.15255E-02
12	.33657E-02	.53045E-02	.31757E-02	.32315E-03	-.14940E-02	.18556E-02	.12121E-02
13	.25379E-02	.43107E-02	.23692E-02	.35970E-03	-.97139E-03	.12435E-02	.83963E-03
14	.18214E-02	.27998E-02	.15537E-02	.25607E-03	-.41925E-03	.43220E-03	.46672E-03
15	.15360E-02	.16090E-02	.11016E-02	-.11297E-03	.64822E-04	-.21220E-03	.25951E-03
16	.25432E-02	.13454E-02	.13799E-02	.65363E-03	.34006E-03	-.82983E-03	.36464E-03
17	.84603E-02	.36529E-02	.26345E-02	.16464E-03	.17788E-03	-.13094E-02	.17655E-02
18	.45300E-02	.17600E-02	.20000E-02	.10837E-03	.11600E-03	-.15800E-02	.70718E-03
19	.10036E-02	.36773E-03	.54345E-03	.37861E-04	-.94357E-05	-.33599E-03	.83577E-04
20	.26165E-03	.10967E-03	.12341E-03	-.11398E-03	.66633E-05	-.88449E-04	.10593E-04
21	.11001E-03	.49834E-04	.42219E-04	0.	.34727E-05	-.23651E-04	.27943E-05
22	.53736E-04	.32591E-04	.13758E-04	0.	.20516E-05	-.48516E-05	.93930E-06
23	.48140E-04	.27930E-04	.10795E-04	0.	.48378E-06	-.31783E-05	.75909E-06
24	.42477E-04	.26808E-04	.10106E-04	0.	.37692E-06	-.23769E-05	.66457E-06
25	.37278E-04	.26000E-04	.86639E-05	0.	.27047E-06	-.17114E-05	.57340E-06
26	.33672E-04	.26000E-04	.56245E-05	0.	.17558E-06	-.14268E-05	.49906E-06
27	.29746E-04	.26000E-04	.23149E-05	0.	.72267E-07	-.11168E-05	.41812E-06
28	.25188E-04	.24255E-04	0.	0.	0.	-.83960E-06	.33724E-06
29	.19671E-04	.18943E-04	0.	0.	0.	-.65571E-06	.26338E-06
30	.13664E-04	.13158E-04	0.	0.	0.	-.45547E-06	.18295E-06
31	.71228E-05	.68590E-05	0.	0.	0.	-.23743E-06	.95367E-07
32	.13010E-17	.11926E-17	0.	0.	0.	-.40658E-19	.18635E-19

```
SDAT9
UT      = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
            0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, I, I, I, I, I, I, I, I, I, I, I, I,
          I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I,
VT      = 0.0, 0.45753928726119E-02, 0.95575318695113E-02,              0.15369859530554E-01, 0.23390482653661E-01,
            0.32124128124909F-01, 0.43110702497628E-01, 0.55206055006563F-01,   0.65423018477173E-01, 0.73883420440922E-01,
            0.78573435488818E-01, 0.83737981431547E-01, 0.80485076509386E-01,   0.72495897591702E-01, 0.61688714406455E-01,
            0.45031624825047E-01, 0.20726043935193E-01, -0.22E-01,            -0.33492096560108E-01, -0.52E-02,
          -0.35756997140019E-02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,        0.0, 0.0, 0.0, 0.0, I, I, I, I, I, I, I, I,
          I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I,
WT      = 0.69072595959405E+00, 0.69227914441323E+00, 0.69693869887076E+00,   0.70346259188365E+00, 0.71690028507420E+00,
            0.73153257068047E+00, 0.7618227364586F+00, 0.79639693101871E+00,   0.83463585862802E+00, 0.87608043406254E+00,
            0.92088057780364E+00, 0.97093564075382E+00, 0.10204764522499E+01,   0.10591321261469E+01, 0.10814494889277E+01,
            0.10835480146038E+01, 0.10546054326319E+01, 0.1006E+01,           0.10018135959333E+01, 0.1E+01, 0.1F+01,
            0.1E+01, 0.1E+01, 0.1E+01, 0.1E+01, 0.1F+01, 0.1F+01, 0.1E+01,   0.1E+01, 0.1E+01, 0.1E+01, 0.1F+01, I, I, I,
          I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I,   I, I, I,
SEND
```

```
REVISED PRESSURE CALCULATION, IPRCALC = 1
SWIRL ANGLES(DEGREES)     MAXIMUM =  .4929E+01     CORE EDGE =     IIIII
```

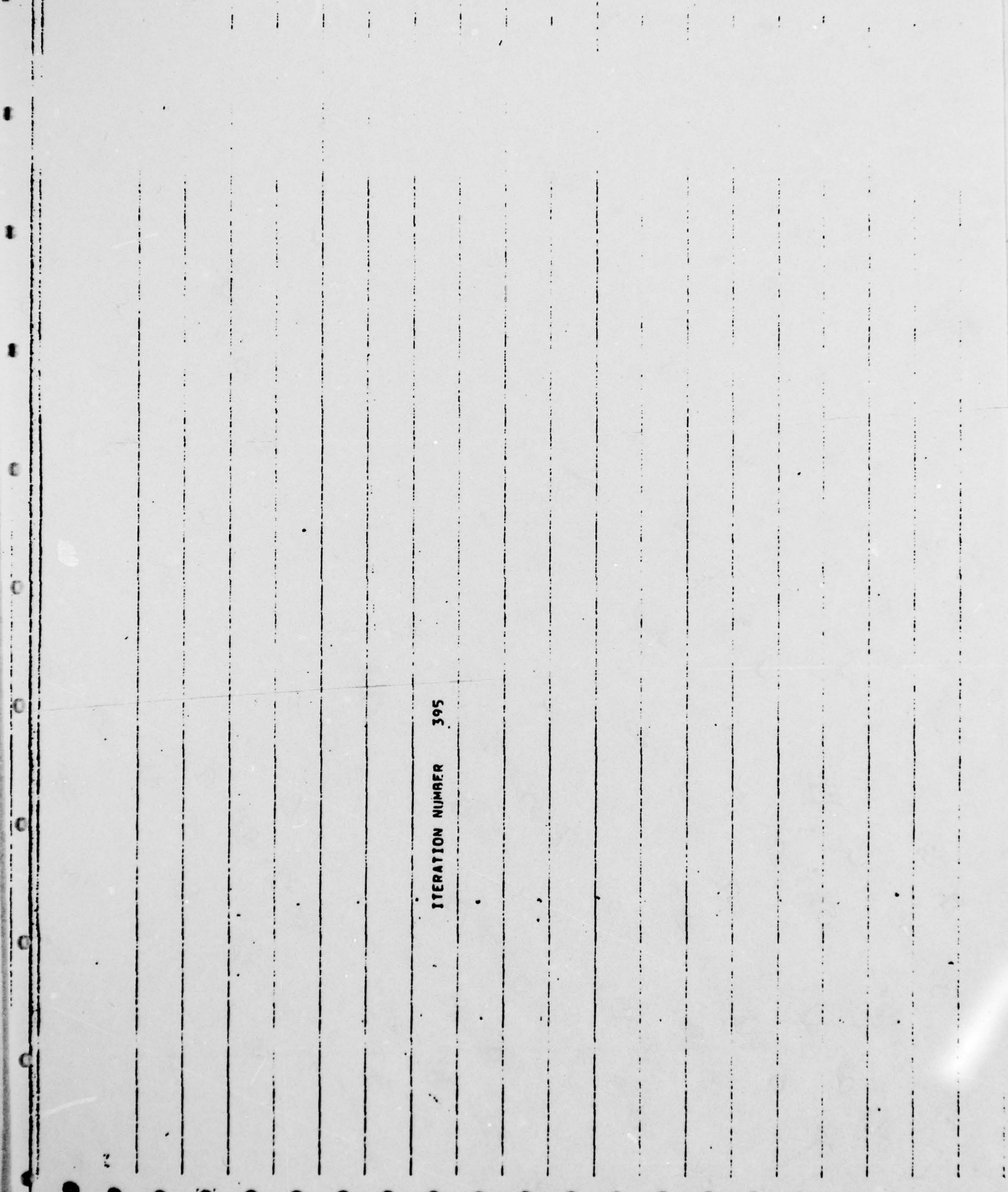
H A N J A L I C   L A U N D E R   S E C O N D   O R D E R   C L O S U R E
T U R B U L E N C E   M O D E L

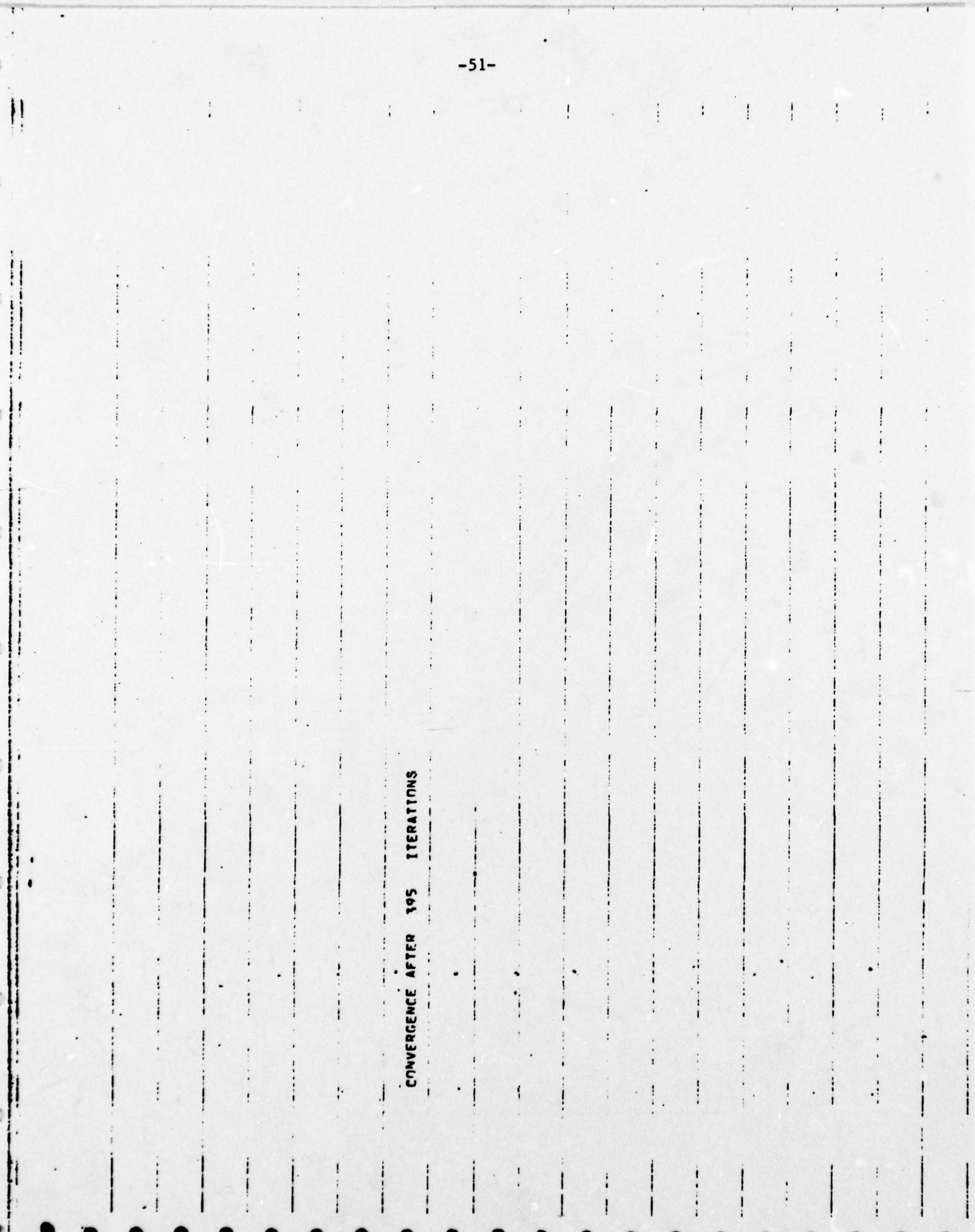
LAUNDER-REECE-RODI PRESSURE-MEAN STRAIN CORRELATION

H A N J A L I C - L A U N D E R   T U R B U L E N T   D I F F U S I O N

```
$DATA8
NHL     =  1,
CEPS    =  0.15E+00,
CEPS1   =  0.140E+01,
CEPS2   =  0.19E+01,
CSN     =  0.11E+00,
CS0     =  0.25E+00,
CS1     =  0.11E+00,
CON1    =  0.15E+01,
CON2    =  0.4E+00,
ISBCTT  =  1,
$END
```

```
MASS FLUX CALCULATION
WEST FLUX =  -.800403E+01      EAST FLUX =   .800588E+01      NORTH FLUX =  -.193655E-02
    NET OUT-FLUX =  -.917840E-04


AFTER  355   TIME STEPS, RMS DIVERGENCE =  .1354E-05
BASED ON THIS AND ITS VALUE AFTER  350   STEPS, THE TIME RATE OF CHANGE OF RMS DIV = -.2145E-07

AFTER  360   TIME STEPS, RMS DIVERGENCE =  .1354E-05
BASED ON THIS AND ITS VALUE AFTER  355   STEPS, THE TIME RATE OF CHANGE OF RMS DIV = -.1556E-07

AFTER  365   TIME STEPS, RMS DIVERGENCE =  .1354E-05
BASED ON THIS AND ITS VALUE AFTER  360   STEPS, THE TIME RATE OF CHANGE OF RMS DIV = -.8917E-08

AFTER  370   TIME STEPS, RMS DIVERGENCE =  .1354E-05
BASED ON THIS AND ITS VALUE AFTER  365   STEPS, THE TIME RATE OF CHANGE OF RMS DIV = -.1350E-07

AFTER  375   TIME STEPS, RMS DIVERGENCE =  .1354E-05
BASED ON THIS AND ITS VALUE AFTER  370   STEPS, THE TIME RATE OF CHANGE OF RMS DIV = -.3206E-08

AFTER  380   TIME STEPS, RMS DIVERGENCE =  .1354E-05
BASED ON THIS AND ITS VALUE AFTER  375   STEPS, THE TIME RATE OF CHANGE OF RMS DIV = -.6131E-08

AFTER  385   TIME STEPS, RMS DIVERGENCE =  .1354E-05
BASED ON THIS AND ITS VALUE AFTER  380   STEPS, THE TIME RATE OF CHANGE OF RMS DIV = -.6016E-08

AFTER  390   TIME STEPS, RMS DIVERGENCE =  .1354E-05
BASED ON THIS AND ITS VALUE AFTER  385   STEPS, THE TIME RATE OF CHANGE OF RMS DIV = -.1726E-08

AFTER  395   TIME STEPS, RMS DIVERGENCE =  .1354E-05
BASED ON THIS AND ITS VALUE AFTER  390   STEPS, THE TIME RATE OF CHANGE OF RMS DIV = -.4711E-09
```


ITERATION NUMBER 395


CONVERGENCE AFTER 395 ITERATIONS

CONVERGENCE ACHIEVED WITH CONCRIT = .1000E-08

J = 1 R(J) = 0.

I	Z	U	V	W	P	DIV
1	.40000E+01	0.	0.	.69073E+00	-.15664E-01	.99245E-04
2	.40748E+01	0.	0.	.69115E+00	-.14366E-01	-.50404E-03
3	.41525E+01	0.	0.	.69225E+00	-.13472E-01	-.28697E-03
4	.42331E+01	0.	0.	.69387E+00	-.12874E-01	-.14892E-03
5	.43168E+01	0.	0.	.69588E+00	-.12471E-01	-.83523E-04
6	.44038E+01	0.	0.	.69816E+00	-.12190E-01	-.41949E-04
7	.44940E+01	0.	0.	.70064E+00	-.11981E-01	-.92826E-05
8	.45878E+01	0.	0.	.70328E+00	-.11812E-01	.12978E-04
9	.46851E+01	0.	0.	.70604E+00	-.11663E-01	.24890E-04
10	.47861E+01	0.	0.	.70890E+00	-.11520E-01	.29461E-04
11	.48911E+01	0.	0.	.71184E+00	-.11377E-01	.29698E-04
12	.50000E+01	0.	0.	.71486E+00	-.11228E-01	.27555E-04
13	.51131E+01	0.	0.	.71794E+00	-.11071E-01	.24224E-04
14	.52306E+01	0.	0.	.72109E+00	-.10906E-01	.20459E-04
15	.53525E+01	0.	0.	.72431E+00	-.10732E-01	.16724E-04
16	.54792E+01	0.	0.	.72759E+00	-.10549E-01	.13268E-04
17	.56107E+01	0.	0.	.73093E+00	-.10359E-01	.10201E-04
18	.57472E+01	0.	0.	.73433E+00	-.10162E-01	.75525E-05
19	.58890E+01	0.	0.	.73779E+00	-.99591E-02	.53091E-05
20	.60361E+01	0.	0.	.74133E+00	-.97517E-02	.34364E-05
21	.61890E+01	0.	0.	.74492E+00	-.95407E-02	.18917E-05
22	.63477E+01	0.	0.	.74858E+00	-.93273E-02	.63096E-06
23	.65125E+01	0.	0.	.75231E+00	-.91127E-02	-.38746E-06
24	.66836E+01	0.	0.	.75611E+00	-.88976E-02	-.12013E-05
25	.68612E+01	0.	0.	.75997E+00	-.86832E-02	-.18436E-05
26	.70457E+01	0.	0.	.76390E+00	-.84702E-02	-.23430E-05
27	.72372E+01	0.	0.	.76790E+00	-.82593E-02	-.27234E-05
28	.74361E+01	0.	0.	.77196E+00	-.80512E-02	-.30045E-05
29	.76426E+01	0.	0.	.77609E+00	-.78465E-02	-.32021E-05
30	.78570E+01	0.	0.	.78028E+00	-.76456E-02	-.33281E-05
31	.80796E+01	0.	0.	.78454E+00	-.74487E-02	-.33913E-05
32	.83107E+01	0.	0.	.78885E+00	-.72560E-02	-.33978E-05
33	.85508E+01	0.	0.	.79322E+00	-.70676E-02	-.33517E-05
34	.88000E+01	0.	0.	.79764E+00	-.68835E-02	-.32558E-05
35	.90588E+01	0.	0.	.80212E+00	-.67034E-02	-.31125E-05
36	.93274E+01	0.	0.	.80663E+00	-.65271E-02	-.29246E-05
37	.96064E+01	0.	0.	.81119E+00	-.63542E-02	-.26960E-05
38	.98961E+01	0.	0.	.81578E+00	-.61844E-02	-.24320E-05
39	.10197E+02	0.	0.	.82040E+00	-.60171E-02	-.21398E-05
40	.10509E+02	0.	0.	.82505E+00	-.58518E-02	-.18281E-05
41	.10833E+02	0.	0.	.82970E+00	-.56880E-02	-.15070E-05
42	.11170E+02	0.	0.	.83437E+00	-.55252E-02	-.11874E-05
43	.11520E+02	0.	0.	.83904E+00	-.53629E-02	-.88011E-06
44	.11883E+02	0.	0.	.84371E+00	-.52006E-02	-.59535E-06
45	.12260E+02	0.	0.	.84837E+00	-.50381E-02	-.34192E-06
46	.12651E+02	0.	0.	.85302E+00	-.48750E-02	-.12681E-06
47	.13058E+02	0.	0.	.85764E+00	-.47111E-02	.153[illegible]E-07
48	.13479E+02	0.	0.	.86224E+00	-.45463E-02	.17175E-06
49	.13918E+02	0.	0.	.86681E+00	-.43805E-02	.25273E-06
50	.14373E+02	0.	0.	.87134E+00	-.42140E-02	[illegible]
51	.14845E+02	0.	0.	.87584E+00	-.40467E-02	[illegible]
52	.15335E+02	0.	0.	.88029E+00	-.38790E-02	[illegible]
53	.15845E+02	0.	0.	.88470E+00	-.37112E-02	[illegible]
54	.16373E+02	0.	0.	.88905E+00	-.35436E-02	.32831E-07
55	.16922E+02	0.	0.	.89336E+00	-.33767E-02	.11911E-06
56	.17493E+02	0.	0.	.89761E+00	-.32107E-02	-.62462E-06
57	.18085E+02	0.	0.	.90180E+00	-.30469E-02	.13356E-05
58	.18699E+02	0.	0.	.90592E+00	-.28821E-02	-.60857E-05

58	.18699E+02	0.	0.	[illegible]	[illegible]	[illegible]	
59	.19337E+02	0.	0.	.91000E+00	-.27[illegible]5E-02	-.80635E-05	
60	.20000E+02	0.	0.	.91404E+00	-.27025E-02	0.	

J = 1 R(J) = 0.

I	TRR	TTT	TZZ	TRT	TRZ	TZT	EPSL
1	.35042E-02	.35042E-02	.25647E-02	0.	0.	0.	.87757E-03
2	.34575E-02	.34623E-02	.25582E-02	0.	0.	0.	.85100E-03
3	.34355E-02	.34444E-02	.25582E-02	0.	0.	0.	.82985E-03
4	.34326E-02	.34448E-02	.25658E-02	0.	0.	0.	.81359E-03
5	.34433E-02	.34580E-02	.25813E-02	0.	0.	0.	.80152E-03
6	.34630E-02	.34796E-02	.26043E-02	0.	0.	0.	.79294E-03
7	.34878E-02	.35055E-02	.26337E-02	0.	0.	0.	.78712E-03
8	.35143E-02	.35328E-02	.26684E-02	0.	0.	0.	.78339E-03
9	.35402E-02	.35590E-02	.27071E-02	0.	0.	0.	.78114E-03
10	.35638E-02	.35825E-02	.27485E-02	0.	0.	0.	.77980E-03
11	.35839E-02	.36024E-02	.27914E-02	0.	0.	0.	.77889E-03
12	.35998E-02	.36179E-02	.28350E-02	0.	0.	0.	.77800E-03
13	.36112E-02	.36288E-02	.28784E-02	0.	0.	0.	.77680E-03
14	.36180E-02	.36349E-02	.29209E-02	0.	0.	0.	.77503E-03
15	.36201E-02	.36365E-02	.29621E-02	0.	0.	0.	.77249E-03
16	.36179E-02	.36335E-02	.30014E-02	0.	0.	0.	.76904E-03
17	.36114E-02	.36264E-02	.30387E-02	0.	0.	0.	.76460E-03
18	.36010E-02	.36154E-02	.30736E-02	0.	0.	0.	.75913E-03
19	.35870E-02	.36008E-02	.31061E-02	0.	0.	0.	.75260E-03
20	.35698E-02	.35828E-02	.31359E-02	0.	0.	0.	.74504E-03
21	.35495E-02	.35620E-02	.31631E-02	0.	0.	0.	.73649E-03
22	.35265E-02	.35384E-02	.31877E-02	0.	0.	0.	.72699E-03
23	.35012E-02	.35125E-02	.32096E-02	0.	0.	0.	.71662E-03
24	.34738E-02	.34846E-02	.32289E-02	0.	0.	0.	.70545E-03
25	.34446E-02	.34548E-02	.32456E-02	0.	0.	0.	.69356E-03
26	.34138E-02	.34235E-02	.32599E-02	0.	0.	0.	.68103E-03
27	.33816E-02	.33909E-02	.32718E-02	0.	0.	0.	.66795E-03
28	.33483E-02	.33571E-02	.32815E-02	0.	0.	0.	.65439E-03
29	.33140E-02	.33224E-02	.32889E-02	0.	0.	0.	.64043E-03
30	.32789E-02	.32870E-02	.32941E-02	0.	0.	0.	.62615E-03
31	.32432E-02	.32508E-02	.32974E-02	0.	0.	0.	.61161E-03
32	.32069E-02	.32141E-02	.32985E-02	0.	0.	0.	.59688E-03
33	.31701E-02	.31770E-02	.32977E-02	0.	0.	0.	.58201E-03
34	.31328E-02	.31394E-02	.32949E-02	0.	0.	0.	.56704E-03
35	.30951E-02	.31013E-02	.32902E-02	0.	0.	0.	.55202E-03
36	.30569E-02	.30628E-02	.32834E-02	0.	0.	0.	.53699E-03
37	.30183E-02	.30239E-02	.32745E-02	0.	0.	0.	.52196E-03
38	.29791E-02	.29845E-02	.32636E-02	0.	0.	0.	.50696E-03
39	.29394E-02	.29445E-02	.32504E-02	0.	0.	0.	.49202E-03
40	.28991E-02	.29039E-02	.32349E-02	0.	0.	0.	.47714E-03
41	.28581E-02	.28627E-02	.32171E-02	0.	0.	0.	.46234E-03
42	.28164E-02	.28207E-02	.31969E-02	0.	0.	0.	.44762E-03
43	.27739E-02	.27779E-02	.31742E-02	0.	0.	0.	.43299E-03
44	.27305E-02	.27343E-02	.31489E-02	0.	0.	0.	.41845E-03
45	.26862E-02	.26898E-02	.31210E-02	0.	0.	0.	.40402E-03
46	.26410E-02	.26443E-02	.30904E-02	0.	0.	0.	.38970E-03
47	.25948E-02	.25979E-02	.30572E-02	0.	0.	0.	.37549E-03
48	.25476E-02	.25505E-02	.30212E-02	0.	0.	0.	.36140E-03
49	.24995E-02	.25021E-02	.29827E-02	0.	0.	0.	.34744E-03
50	.24502E-02	.24527E-02	.29415E-02	0.	0.	0.	.33361E-03
51	.24000E-02	.24022E-02	.28977E-02	0.	0.	0.	.31993E-03
52	.23487E-02	.23508E-02	.28514E-02	0.	0.	0.	.30641E-03
53	.22965E-02	.22983E-02	.28027E-02	0.	0.	0.	.29307E-03
54	.22433E-02	.22449E-02	.27516E-02	0.	0.	0.	.27991E-03
55	.21891E-02	.21905E-02	.26983E-02	0.	0.	0.	.26696E-03
56	.21340E-02	.21353E-02	.26428E-02	0.	0.	0.	.25423E-03
							.24173E-03

59	[illegible]	[illegible]	[illegible]	[illegible]		
60	.[illegible]9062E-02	.19068E-02	.24025E-02	0.	0.	.20584E-[illegible]

J = 2 R(J) = .27321E-01

I	Z	U	V	W	P	DIV
1	.40000E+01	0.	.45754E-02	.69228E+00	-.15605E-01	.20036E-03
2	.40748E+01	-.14821E-03	.46612E-02	.69270E+00	-.14333E-01	-.46779E-03
3	.41525E+01	-.24190E-03	.47562E-02	.69379E+00	-.13451E-01	-.26291E-03
4	.42331E+01	-.30174E-03	.48430E-02	.69540E+00	-.12857E-01	-.14243E-03
5	.43168E+01	-.33935E-03	.49147E-02	.69738E+00	-.12454E-01	-.94487E-04
6	.44038E+01	-.36138E-03	.49691E-02	.69965E+00	-.12169E-01	-.57740E-04
7	.44940E+01	-.37284E-03	.50054E-02	.70211E+00	-.11956E-01	-.23421E-04
8	.45878E+01	-.37750E-03	.50239E-02	.70474E+00	-.11782E-01	.58761E-06
9	.45851E+01	-.37789E-03	.50253E-02	.70748E+00	-.11628E-01	.13776E-04
10	.47861E+01	-.37555E-03	.50106E-02	.71032E+00	-.11482E-01	.19949E-04
11	.48911E+01	-.37147E-03	.49810E-02	.71325E+00	-.11335E-01	.22101E-04
12	.50000E+01	-.36633E-03	.49377E-02	.71625E+00	-.11184E-01	.21865E-04
13	.51131E+01	-.36054E-03	.48819E-02	.71932E+00	-.11026E-01	.20244E-04
14	.52306E+01	-.35439E-03	.48148E-02	.72246E+00	-.10861E-01	.17931E-04
15	.53525E+01	-.34808E-03	.47378E-02	.72566E+00	-.10687E-01	.15373E-04
16	.54792E+01	-.34171E-03	.46521E-02	.72892E+00	-.10505E-01	.12832E-04
17	.56107E+01	-.33536E-03	.45588E-02	.73225E+00	-.10315E-01	.10448E-04
18	.57472E+01	-.32906E-03	.44589E-02	.73564E+00	-.10119E-01	.82895E-05
19	.58890E+01	-.32283E-03	.43538E-02	.73909E+00	-.99182E-02	.63824E-05
20	.60361E+01	-.31668E-03	.42442E-02	.74260E+00	-.97125E-02	.47256E-05
21	.61890E+01	-.31061E-03	.41313E-02	.74618E+00	-.95035E-02	.33049E-05
22	.63477E+01	-.30461E-03	.40159E-02	.74982E+00	-.92921E-02	.20991E-05
23	.65125E+01	-.29868E-03	.38990E-02	.75353E+00	-.90794E-02	.10848E-05
24	.66836E+01	-.29279E-03	.37814E-02	.75730E+00	-.88663E-02	.23826E-06
25	.68612E+01	-.28693E-03	.36640E-02	.76115E+00	-.86539E-02	-.46275E-06
26	.70457E+01	-.28110E-03	.35474E-02	.76505E+00	-.84428E-02	-.10384E-05
27	.72372E+01	-.27528E-03	.34325E-02	.76903E+00	-.82338E-02	-.15061E-05
28	.74361E+01	-.26945E-03	.33199E-02	.77307E+00	-.80275E-02	-.18806E-05
29	.76426E+01	-.26360E-03	.32102E-02	.77717E+00	-.78244E-02	-.21734E-05
30	.78570E+01	-.25771E-03	.31041E-02	.78134E+00	-.76251E-02	-.23934E-05
31	.80796E+01	-.25177E-03	.30021E-02	.78557E+00	-.74296E-02	-.25471E-05
32	.83107E+01	-.24578E-03	.29047E-02	.78986E+00	-.72384E-02	-.26389E-05
33	.85508E+01	-.23971E-03	.28124E-02	.79421E+00	-.70512E-02	-.26720E-05
34	.88000E+01	-.23356E-03	.27254E-02	.79861E+00	-.68683E-02	-.26488E-05
35	.90588E+01	-.22732E-03	.26441E-02	.80305E+00	-.66892E-02	-.25723E-05
36	.93274E+01	-.22099E-03	.25686E-02	.80755E+00	-.65139E-02	-.24457E-05
37	.96064E+01	-.21457E-03	.24990E-02	.81208E+00	-.63419E-02	-.22740E-05
38	.98961E+01	-.20807E-03	.24354E-02	.81665E+00	-.61729E-02	-.20635E-05
39	.10197E+02	-.20150E-03	.23775E-02	.82124E+00	-.60063E-02	-.18219E-05
40	.10509E+02	-.19487E-03	.23253E-02	.82586E+00	-.58416E-02	-.15583E-05
41	.10833E+02	-.18820E-03	.22784E-02	.83049E+00	-.56784E-02	-.12828E-05
42	.11170E+02	-.18150E-03	.22365E-02	.83514E+00	-.55161E-02	-.10054E-05
43	.11520E+02	-.17481E-03	.21989E-02	.83978E+00	-.53543E-02	-.73612E-06
44	.11883E+02	-.16815E-03	.21654E-02	.84443E+00	-.51925E-02	-.48371E-06
45	.12260E+02	-.16153E-03	.21352E-02	.84907E+00	-.50304E-02	-.25569E-06
46	.12651E+02	-.15498E-03	.21078E-02	.85369E+00	-.48678E-02	-.57782E-07
47	.13058E+02	-.14852E-03	.20826E-02	.85829E+00	-.47043E-02	.10622E-06
48	.13479E+02	-.14217E-03	.20590E-02	.86287E+00	-.45399E-02	.23427E-06
49	.13918E+02	-.13595E-03	.20363E-02	.86742E+00	-.43745E-02	.32658E-06
50	.14373E+02	-.12987E-03	.20141E-02	.87193E+00	-.42084E-02	.38335E-06
51	.14845E+02	-.12395E-03	.19918E-02	.87640E+00	-.40415E-02	.41072E-06
52	.15335E+02	-.11818E-03	.19689E-02	.88084E+00	-.38743E-02	.40115E-06
53	.15845E+02	-.11259E-03	.19450E-02	.88522E+00	-.37069E-02	.39270E-06
54	.16373E+02	-.10716E-03	.19196E-02	.88956E+00	-.35398E-02	.28530E-06
55	.16922E+02	-.10192E-03	.18924E-02	.89384E+00	-.33733E-02	.40851E-06
56	.17493E+02	-.96830E-04	.18630E-02	.89807E+00	-.32078E-02	-.25947E-06
57	.18085E+02	-.91996E-04	.18313E-02	.90224E+00	-.30444E-02	.16817E-05
58	.18699E+02	-.87012E-04	.17968E-02	.90635E+00	-.28801E-02	-.54923E-05

J = 2 R(J) = .27321E-01

I	TRR	TTT	TZZ	TRT	TRZ	TZT	EPSL
1	.35066E-02	.35066E-02	.25730E-02	-.52112E-06	-.28045E-03	.65833E-04	.87938E-03
2	.34689E-02	.34751E-02	.25723E-02	-.49581E-05	-.28355E-03	.74061E-04	.85587E-03
3	.34529E-02	.34662E-02	.25780E-02	-.55460E-05	-.28586E-03	.82289E-04	.83733E-03
4	.34535E-02	.34738E-02	.25908E-02	-.47987E-05	-.28816E-03	.90092E-04	.82312E-03
5	.34660E-02	.34927E-02	.26107E-02	-.36547E-05	-.29043E-03	.97347E-04	.81260E-03
6	.34860E-02	.35183E-02	.26371E-02	-.24555E-05	-.29237E-03	.10402E-03	.80512E-03
7	.35100E-02	.35470E-02	.26691E-02	-.13222E-05	-.29368E-03	.11008E-03	.80002E-03
8	.35353E-02	.35759E-02	.27056E-02	-.29606E-06	-.29418E-03	.11553E-03	.79671E-03
9	.35597E-02	.36029E-02	.27454E-02	.61061E-06	-.29382E-03	.12039E-03	.79463E-03
10	.35816E-02	.36265E-02	.27875E-02	.13971E-05	-.29266E-03	.12467E-03	.79326E-03
11	.36001E-02	.36459E-02	.28306E-02	.20681E-05	-.29081E-03	.12842E-03	.79217E-03
12	.36145E-02	.36605E-02	.28741E-02	.26311E-05	-.28839E-03	.13166E-03	.79099E-03
13	.36245E-02	.36703E-02	.29172E-02	.30948E-05	-.28551E-03	.13443E-03	.78941E-03
14	.36300E-02	.36751E-02	.29593E-02	.34687E-05	-.28229E-03	.13676E-03	.78720E-03
15	.36311E-02	.36752E-02	.29998E-02	.37618E-05	-.27880E-03	.13870E-03	.78418E-03
16	.36279E-02	.36707E-02	.30385E-02	.39830E-05	-.27512E-03	.14025E-03	.78024E-03
17	.36207E-02	.36620E-02	.30751E-02	.41404E-05	-.27128E-03	.14145E-03	.77530E-03
18	.36096E-02	.36494E-02	.31092E-02	.42413E-05	-.26733E-03	.14231E-03	.76932E-03
19	.35951E-02	.36332E-02	.31408E-02	.42926E-05	-.26331E-03	.14285E-03	.76230E-03
20	.35773E-02	.36137E-02	.31699E-02	.43005E-05	-.25922E-03	.14310E-03	.75425E-03
21	.35566E-02	.35913E-02	.31962E-02	.42704E-05	-.25508E-03	.14307E-03	.74523E-03
22	.35333E-02	.35663E-02	.32199E-02	.42074E-05	-.25092E-03	.14277E-03	.73527E-03
23	.35076E-02	.35390E-02	.32410E-02	.41159E-05	-.24672E-03	.14223E-03	.72447E-03
24	.34799E-02	.35097E-02	.32594E-02	.40001E-05	-.24251E-03	.14147E-03	.71288E-03
25	.34504E-02	.34787E-02	.32754E-02	.38637E-05	-.23827E-03	.14050E-03	.70059E-03
26	.34194E-02	.34462E-02	.32888E-02	.37100E-05	-.23402E-03	.13935E-03	.68768E-03
27	.33870E-02	.34124E-02	.32999E-02	.35420E-05	-.22975E-03	.13803E-03	.67423E-03
28	.33535E-02	.33775E-02	.33087E-02	.33628E-05	-.22547E-03	.13658E-03	.66033E-03
29	.33191E-02	.33418E-02	.33153E-02	.31747E-05	-.22115E-03	.13501E-03	.64604E-03
30	.32839E-02	.33053E-02	.33198E-02	.29804E-05	-.21681E-03	.13334E-03	.63145E-03
31	.32480E-02	.32683E-02	.33222E-02	.27822E-05	-.21243E-03	.13159E-03	.61662E-03
32	.32115E-02	.32307E-02	.33226E-02	.25822E-05	-.20801E-03	.12978E-03	.60161E-03
33	.31746E-02	.31927E-02	.33210E-02	.23826E-05	-.20354E-03	.12793E-03	.58646E-03
34	.31371E-02	.31542E-02	.33175E-02	.21855E-05	-.19901E-03	.12605E-03	.57124E-03
35	.30993E-02	.31154E-02	.33119E-02	.19929E-05	-.19441E-03	.12415E-03	.55598E-03
36	.30609E-02	.30762E-02	.33044E-02	.18067E-05	-.18975E-03	.12224E-03	.54071E-03
37	.30222E-02	.30366E-02	.32948E-02	.16288E-05	-.18501E-03	.12032E-03	.52546E-03
38	.29829E-02	.29965E-02	.32830E-02	.14606E-05	-.18020E-03	.11840E-03	.51025E-03
39	.29430E-02	.29559E-02	.32691E-02	.13037E-05	-.17532E-03	.11647E-03	.49510E-03
40	.29025E-02	.29147E-02	.32530E-02	.11592E-05	-.17037E-03	.11453E-03	.48002E-03
41	.28614E-02	.28728E-02	.32344E-02	.10279E-05	-.16537E-03	.11258E-03	.46503E-03
42	.28195E-02	.28302E-02	.32135E-02	.91033E-06	-.16032E-03	.11059E-03	.45013E-03
43	.27768E-02	.27869E-02	.31901E-02	.80680E-06	-.15523E-03	.10857E-03	.43533E-03
44	.27333E-02	.27427E-02	.31641E-02	.71714E-06	-.15012E-03	.10651E-03	.42064E-03
45	.26888E-02	.26977E-02	.31355E-02	.64093E-06	-.14500E-03	.10439E-03	.40605E-03
46	.26435E-02	.26517E-02	.31042E-02	.57751E-06	-.13989E-03	.10221E-03	.39158E-03
47	.25972E-02	.26047E-02	.30703E-02	.52598E-06	-.13480E-03	.99961E-04	.37723E-03
48	.25498E-02	.25568E-02	.30337E-02	.48532E-06	-.12975E-03	.97637E-04	.36300E-03
49	.25015E-02	.25079E-02	.29945E-02	.45440E-06	-.12475E-03	.95234E-04	.34891E-03
50	.24522E-02	.24580E-02	.29527E-02	.43203E-06	-.11982E-03	.92751E-04	.33497E-03
51	.24018E-02	.24070E-02	.29082E-02	.41704E-06	-.11495E-03	.90186E-04	.32117E-03
52	.23504E-02	.23551E-02	.28613E-02	.40830E-06	-.11018E-03	.87541E-04	.30755E-03
53	.22981E-02	.23022E-02	.28120E-02	.40474E-06	-.10549E-03	.84819E-04	.29410E-03
54	.22448E-02	.22483E-02	.27603E-02	.40537E-06	-.10090E-03	.82024E-04	.28085E-03
55	.21905E-02	.21935E-02	.27064E-02	.40930E-06	-.96418E-04	.79161E-04	.26781E-03
56	.21353E-02	.21378E-02	.26500E-02	.41574E-06	-.92042E-04	.76236E-04	.25499E-03
57	.20794E-02	.20813E-02	.25924E-02	.42398E-06	-.87777E-04	.73259E-04	.24242E-03
58	.20226E-02	.20241E-02	.25326E-02	.43341E-06	-.83625E-04	.70235E-04	.23011E-03

60 .19072E-02 .19078E-02 .24080E-02 .45138E-06 [illegible]

J = 3 R(J) = .57070E-01

I	Z	U	V	W	P	DIV
1	.40000E+01	0.	.95575E-02	.69694E+00	-.15500E-01	.31454E-03
2	.40748E+01	-.30682E-03	.97758E-02	.69736E+00	-.14286E-01	-.23642E-04
3	.41525E+01	-.50158E-03	.99750E-02	.69841E+00	-.13432E-01	-.55404E-04
4	.42331E+01	-.62772E-03	.10147E-01	.69997E+00	-.12850E-01	-.10646E-03
5	.43168E+01	-.70790E-03	.10287E-01	.70190E+00	-.12447E-01	-.84514E-04
6	.44038E+01	-.75503E-03	.10395E-01	.70411E+00	-.12158E-01	-.43790E-04
7	.44940E+01	-.77969E-03	.10468E-01	.70653E+00	-.11937E-01	-.12725E-04
8	.45878E+01	-.79004E-03	.10507E-01	.70910E+00	-.11753E-01	.40887E-05
9	.46851E+01	-.79138E-03	.10514E-01	.71180E+00	-.11590E-01	.12656E-04
10	.47861E+01	-.78691E-03	.10488E-01	.71460E+00	-.11435E-01	.16819E-04
11	.48911E+01	-.77870E-03	.10433E-01	.71748E+00	-.11281E-01	.18284E-04
12	.50000E+01	-.76811E-03	.10350E-01	.72044E+00	-.11125E-01	.18048E-04
13	.51131E+01	-.75609E-03	.10241E-01	.72347E+00	-.10963E-01	.16826E-04
14	.52306E+01	-.74324E-03	.10110E-01	.72656E+00	-.10795E-01	.15096E-04
15	.53525E+01	-.72997E-03	.99580E-02	.72972E+00	-.10620E-01	.13153E-04
16	.54792E+01	-.71654E-03	.97878E-02	.73294E+00	-.10437E-01	.11183E-04
17	.56107E+01	-.70310E-03	.96017E-02	.73622E+00	-.10249E-01	.92924E-05
18	.57472E+01	-.68976E-03	.94020E-02	.73956E+00	-.10055E-01	.75433E-05
19	.58890E+01	-.67656E-03	.91908E-02	.74296E+00	-.98555E-02	.59639E-05
20	.60361E+01	-.66354E-03	.89704E-02	.74642E+00	-.96523E-02	.45629E-05
21	.61890E+01	-.65068E-03	.87426E-02	.74994E+00	-.94461E-02	.33369E-05
22	.63477E+01	-.63798E-03	.85095E-02	.75353E+00	-.92377E-02	.22753E-05
23	.65125E+01	-.62543E-03	.82728E-02	.75718E+00	-.90282E-02	.13638E-05
24	.66836E+01	-.61300E-03	.80343E-02	.76090E+00	-.88184E-02	.58682E-06
25	.68612E+01	-.60066E-03	.77958E-02	.76467E+00	-.86092E-02	-.71156E-07
26	.70457E+01	-.58839E-03	.75588E-02	.76852E+00	-.84013E-02	-.62454E-06
27	.72372E+01	-.57615E-03	.73247E-02	.77242E+00	-.81954E-02	-.10859E-05
28	.74361E+01	-.56392E-03	.70951E-02	.77639E+00	-.79921E-02	-.14657E-05
29	.76426E+01	-.55165E-03	.68712E-02	.78043E+00	-.77919E-02	-.17719E-05
30	.78570E+01	-.53933E-03	.66543E-02	.78453E+00	-.75952E-02	-.20104E-05
31	.80796E+01	-.52692E-03	.64456E-02	.78868E+00	-.74022E-02	-.21853E-05
32	.83107E+01	-.51438E-03	.62459E-02	.79290E+00	-.72132E-02	-.22992E-05
33	.85508E+01	-.50170E-03	.60563E-02	.79717E+00	-.70282E-02	-.23541E-05
34	.88000E+01	-.48886E-03	.58773E-02	.80149E+00	-.68471E-02	-.23521E-05
35	.90588E+01	-.47583E-03	.57097E-02	.80587E+00	-.66698E-02	-.22959E-05
36	.93274E+01	-.46263E-03	.55538E-02	.81028E+00	-.64960E-02	-.21894E-05
37	.96064E+01	-.44923E-03	.54097E-02	.81474E+00	-.63253E-02	-.20376E-05
38	.98961E+01	-.43567E-03	.52775E-02	.81923E+00	-.61574E-02	-.18480E-05
39	.10197E+02	-.42196E-03	.51569E-02	.82375E+00	-.59918E-02	-.16283E-05
40	.10509E+02	-.40812E-03	.50475E-02	.82830E+00	-.58280E-02	-.13877E-05
41	.10833E+02	-.39419E-03	.49487E-02	.83286E+00	-.56655E-02	-.11360E-05
42	.11170E+02	-.38022E-03	.48597E-02	.83743E+00	-.55039E-02	-.88280E-06
43	.11520E+02	-.36624E-03	.47796E-02	.84201E+00	-.53427E-02	-.63704E-06
44	.11883E+02	-.35232E-03	.47073E-02	.84658E+00	-.51814E-02	-.40662E-06
45	.12260E+02	-.33849E-03	.46417E-02	.85115E+00	-.50198E-02	-.19795E-06
46	.12651E+02	-.32479E-03	.45815E-02	.85571E+00	-.48576E-02	-.15824E-07
47	.13058E+02	-.31129E-03	.45256E-02	.86024E+00	-.46945E-02	.13674E-06
48	.13479E+02	-.29801E-03	.44727E-02	.86475E+00	-.45306E-02	.25825E-06
49	.13918E+02	-.28499E-03	.44217E-02	.86924E+00	-.43657E-02	.34909E-06
50	.14373E+02	-.27227E-03	.43713E-02	.87369E+00	-.42001E-02	.40957E-06
51	.14845E+02	-.25986E-03	.43206E-02	.87810E+00	-.40338E-02	.44507E-06
52	.15335E+02	-.24778E-03	.42685E-02	.88247E+00	-.38670E-02	.44876E-06
53	.15845E+02	-.23606E-03	.42141E-02	.88680E+00	-.37002E-02	.45443E-06
54	.16373E+02	-.22469E-03	.41566E-02	.89107E+00	-.35337E-02	.37111E-06
55	.16922E+02	-.21371E-03	.40953E-02	.89530E+00	-.33678E-02	.49985E-06
56	.17493E+02	-.20303E-03	.40295E-02	.89947E+00	-.32030E-02	-.10608E-06
57	.18085E+02	-.19289E-03	.39587E-02	.90359E+00	-.30402E-02	.17429E-05
58	.18699E+02	-.18245E-03	.38823E-02	.90764E+00	-.28765E-02	-.50785E-05

59	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		
60	.12000000E+02	-.15660E-03	.37049E-02	.91560E+00	-.26973E-02	0.	

J = 3 R(J) = .57070E-01

I, TRR,	TTT,	T22,	TRT,	TR7,	T7T,	EPSL	
1	.35138E-02	.35138E-02	.25981E-02	-.20946E-04	-.58583E-03	.13752E-03	.88481E-03
2	.35030E-02	.35134E-02	.26148E-02	-.27008E-04	-.58740E-03	.15590E-03	.87048E-03
3	.35052E-02	.35315E-02	.26377E-02	-.25308E-04	-.59108E-03	.17302E-03	.85975E-03
4	.35165E-02	.35611E-02	.26659E-02	-.20937E-04	-.59615E-03	.18917E-03	.85172E-03
5	.35340E-02	.35967E-02	.26988E-02	-.15855E-04	-.60153E-03	.20435E-03	.84584E-03
6	.35548E-02	.36346E-02	.27356E-02	-.10796E-04	-.60614E-03	.21845E-03	.84165E-03
7	.35769E-02	.36715E-02	.27753E-02	-.60384E-05	-.60926E-03	.23137E-03	.83874E-03
8	.35984E-02	.37053E-02	.28172E-02	-.16964E-05	-.61053E-03	.24306E-03	.83668E-03
9	.36181E-02	.37346E-02	.28605E-02	.21828E-05	-.60993E-03	.25351E-03	.83519E-03
10	.36350E-02	.37584E-02	.29044E-02	.55848E-05	-.60758E-03	.26276E-03	.83365E-03
11	.36486E-02	.37764E-02	.29482E-02	.85164E-05	-.60374E-03	.27086E-03	.83201E-03
12	.36585E-02	.37885E-02	.29915E-02	.10998E-04	-.59867E-03	.27787E-03	.82994E-03
13	.36644E-02	.37948E-02	.30337E-02	.13060E-04	-.59263E-03	.28386E-03	.82723E-03
14	.36663E-02	.37956E-02	.30744E-02	.14735E-04	-.58583E-03	.28890E-03	.82371E-03
15	.36641E-02	.37913E-02	.31132E-02	.16060E-04	-.57846E-03	.29305E-03	.81927E-03
16	.36582E-02	.37822E-02	.31499E-02	.17071E-04	-.57066E-03	.29636E-03	.81385E-03
17	.36485E-02	.37687E-02	.31842E-02	.17801E-04	-.56255E-03	.29889E-03	.80739E-03
18	.36354E-02	.37513E-02	.32160E-02	.18283E-04	-.55420E-03	.30068E-03	.79990E-03
19	.36191E-02	.37304E-02	.32452E-02	.18545E-04	-.54570E-03	.30178E-03	.79138E-03
20	.35998E-02	.37062E-02	.32718E-02	.18614E-04	-.53707E-03	.30224E-03	.78188E-03
21	.35778E-02	.36793E-02	.32956E-02	.18515E-04	-.52836E-03	.30208E-03	.77143E-03
22	.35534E-02	.36500E-02	.33168E-02	.18269E-04	-.51960E-03	.30136E-03	.76012E-03
23	.35268E-02	.36185E-02	.33353E-02	.17897E-04	-.51080E-03	.30011E-03	.74800E-03
24	.34982E-02	.35852E-02	.33512E-02	.17417E-04	-.50198E-03	.29839E-03	.73515E-03
25	.34680E-02	.35503E-02	.33645E-02	.16844E-04	-.49313E-03	.29623E-03	.72167E-03
26	.34363E-02	.35141E-02	.33755E-02	.16194E-04	-.48427E-03	.29369E-03	.70761E-03
27	.34033E-02	.34768E-02	.33841E-02	.15481E-04	-.47539E-03	.29081E-03	.69308E-03
28	.33693E-02	.34387E-02	.33904E-02	.14716E-04	-.46648E-03	.28763E-03	.67814E-03
29	.33344E-02	.33998E-02	.33946E-02	.13913E-04	-.45754E-03	.28421E-03	.66288E-03
30	.32987E-02	.33604E-02	.33967E-02	.13080E-04	-.44855E-03	.28058E-03	.64735E-03
31	.32624E-02	.33205E-02	.33967E-02	.12229E-04	-.43949E-03	.27680E-03	.63163E-03
32	.32254E-02	.32803E-02	.33948E-02	.11369E-04	-.43036E-03	.27289E-03	.61577E-03
33	.31880E-02	.32397E-02	.33908E-02	.10510E-04	-.42112E-03	.26890E-03	.59983E-03
34	.31502E-02	.31989E-02	.33850E-02	.96615E-05	-.41177E-03	.26485E-03	.58384E-03
35	.31119E-02	.31578E-02	.33771E-02	.88312E-05	-.40229E-03	.26076E-03	.56784E-03
36	.30731E-02	.31164E-02	.33673E-02	.80279E-05	-.39267E-03	.25665E-03	.55187E-03
37	.30338E-02	.30746E-02	.33555E-02	.72594E-05	-.38291E-03	.25253E-03	.53595E-03
38	.29941E-02	.30325E-02	.33415E-02	.65329E-05	-.37299E-03	.24841E-03	.52010E-03
39	.29537E-02	.29899E-02	.33254E-02	.58544E-05	-.36293E-03	.24427E-03	.50434E-03
40	.29128E-02	.29469E-02	.33070E-02	.52289E-05	-.35273E-03	.24011E-03	.48867E-03
41	.28711E-02	.29032E-02	.32864E-02	.46602E-05	-.34240E-03	.23591E-03	.47312E-03
42	.28288E-02	.28589E-02	.32633E-02	.41507E-05	-.33198E-03	.23166E-03	.45769E-03
43	.27856E-02	.28138E-02	.32377E-02	.37012E-05	-.32147E-03	.22734E-03	.44237E-03
44	.27416E-02	.27680E-02	.32096E-02	.33113E-05	-.31092E-03	.22293E-03	.42719E-03
45	.26967E-02	.27213E-02	.31789E-02	.29791E-05	-.30035E-03	.21841E-03	.41213E-03
46	.26509E-02	.26737E-02	.31457E-02	.27018E-05	-.28979E-03	.21377E-03	.39721E-03
47	.26041E-02	.26252E-02	.31097E-02	.24755E-05	-.27927E-03	.20898E-03	.38244E-03
48	.25564E-02	.25757E-02	.30711E-02	.22959E-05	-.26883E-03	.20405E-03	.36781E-03
49	.25077E-02	.25253E-02	.30300E-02	.21581E-05	-.25848E-03	.19895E-03	.35334E-03
50	.24580E-02	.24739E-02	.29862E-02	.20570E-05	-.24827E-03	.19369E-03	.33903E-03
51	.24072E-02	.24215E-02	.29398E-02	.19875E-05	-.23820E-03	.18826E-03	.32490E-03
52	.23555E-02	.23681E-02	.28911E-02	.19447E-05	-.22831E-03	.18267E-03	.31095E-03
53	.23029E-02	.23137E-02	.28390E-02	.19242E-05	-.21860E-03	.17692E-03	.29720E-03
54	.22492E-02	.22585E-02	.27864E-02	.19215E-05	-.20910E-03	.17102E-03	.28366E-03
55	.21947E-02	.22023E-02	.27307E-02	.19330E-05	-.19980E-03	.16499E-03	.27036E-03
56	.21393E-02	.21454E-02	.26730E-02	.19551E-05	-.19073E-03	.15884E-03	.25729E-03
57	.20830E-02	.20877E-02	.26134E-02	[illegible]	[illegible]	[illegible]	[illegible]

59	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
60	.19159E-02	.19149E-02	.24543E-02	.527?6E-05	-.24342E-03	.20713E-03

.21054E-03

J = 5 R(J) = .12474E+00

I	Z	U	V	W	P	DIV
1	.40000E+01	0.	.23390E-01	.71690E+00	-.15305E-01	.53803E-03
2	.40748E+01	-.60090E-03	.23612E-01	.71726E+00	-.14214E-01	-.18272E-03
3	.41525E+01	-.10345E-02	.23825E-01	.71824E+00	-.13435E-01	-.93038E-04
4	.42331E+01	-.13184E-02	.24014E-01	.71970E+00	-.12887E-01	-.61483E-05
5	.43168E+01	-.14900E-02	.24167E-01	.72150E+00	-.12484E-01	.26968E-04
6	.44038E+01	-.15886E-02	.24277E-01	.72354E+00	-.12171E-01	.30006E-04
7	.44940E+01	-.16419E-02	.24338E-01	.72576E+00	-.11915E-01	.25626E-04
8	.45878E+01	-.16663E-02	.24350E-01	.72812E+00	-.11694E-01	.21040E-04
9	.46851E+01	-.16718E-02	.24309E-01	.73060E+00	-.11495E-01	.17218E-04
10	.47861E+01	-.16647E-02	.24216E-01	.73317E+00	-.11309E-01	.14241E-04
11	.48911E+01	-.16493E-02	.24073E-01	.73582E+00	-.11129E-01	.12017E-04
12	.50000E+01	-.16284E-02	.23881E-01	.73855E+00	-.10952E-01	.10319E-04
13	.51131E+01	-.16040E-02	.23642E-01	.74135E+00	-.10775E-01	.89474E-05
14	.52306E+01	-.15773E-02	.23361E-01	.74421E+00	-.10596E-01	.77725E-05
15	.53525E+01	-.15493E-02	.23040E-01	.74713E+00	-.10415E-01	.67172E-05
16	.54792E+01	-.15207E-02	.22683E-01	.75010E+00	-.10231E-01	.57388E-05
17	.56107E+01	-.14918E-02	.22294E-01	.75314E+00	-.10043E-01	.48182E-05
18	.57472E+01	-.14629E-02	.21877E-01	.75623E+00	-.98525E-02	.39502E-05
19	.58890E+01	-.14342E-02	.21436E-01	.75937E+00	-.96592E-02	.31355E-05
20	.60361E+01	-.14059E-02	.20975E-01	.76257E+00	-.94636E-02	.23766E-05
21	.61890E+01	-.13780E-02	.20497E-01	.76582E+00	-.92662E-02	.16758E-05
22	.63477E+01	-.13504E-02	.20008E-01	.76913E+00	-.90676E-02	.10341E-05
23	.65125E+01	-.13233E-02	.19509E-01	.77250E+00	-.88685E-02	.45112E-06
24	.66836E+01	-.12965E-02	.19005E-01	.77593E+00	-.86694E-02	-.74145E-07
25	.68612E+01	-.12700E-02	.18499E-01	.77941E+00	-.84710E-02	-.54322E-06
26	.70457E+01	-.12438E-02	.17995E-01	.78296E+00	-.82739E-02	-.95741E-06
27	.72372E+01	-.12178E-02	.17496E-01	.78656E+00	-.80784E-02	-.13175E-05
28	.74361E+01	-.11919E-02	.17004E-01	.79022E+00	-.78852E-02	-.16237E-05
29	.76426E+01	-.11660E-02	.16523E-01	.79394E+00	-.76946E-02	-.18754E-05
30	.78570E+01	-.11401E-02	.16055E-01	.79772E+00	-.75070E-02	-.20720E-05
31	.80796E+01	-.11141E-02	.15602E-01	.80156E+00	-.73224E-02	-.22125E-05
32	.83107E+01	-.10879E-02	.15167E-01	.80546E+00	-.71410E-02	-.22968E-05
33	.85508E+01	-.10614E-02	.14752E-01	.80941E+00	-.69629E-02	-.23256E-05
34	.88000E+01	-.10346E-02	.14357E-01	.81341E+00	-.67880E-02	-.23008E-05
35	.90588E+01	-.10074E-02	.13985E-01	.81746E+00	-.66161E-02	-.22261E-05
36	.93274E+01	-.97992E-03	.13635E-01	.82156E+00	-.64470E-02	-.21064E-05
37	.96064E+01	-.95202E-03	.13309E-01	.82570E+00	-.62803E-02	-.19483E-05
38	.98961E+01	-.92375E-03	.13006E-01	.82987E+00	-.61158E-02	-.17593E-05
39	.10197E+02	-.89515E-03	.12725E-01	.83408E+00	-.59530E-02	-.15477E-05
40	.10509E+02	-.86627E-03	.12466E-01	.83831E+00	-.57914E-02	-.13223E-05
41	.10833E+02	-.83718E-03	.12228E-01	.84256E+00	-.56308E-02	-.10914E-05
42	.11170E+02	-.80796E-03	.12008E-01	.84683E+00	-.54706E-02	-.86313E-06
43	.11520E+02	-.77871E-03	.11804E-01	.85111E+00	-.53105E-02	-.64446E-06
44	.11883E+02	-.74951E-03	.11616E-01	.85539E+00	-.51502E-02	-.44138E-06
45	.12260E+02	-.72047E-03	.11439E-01	.85967E+00	-.49894E-02	-.25857E-06
46	.12651E+02	-.69168E-03	.11272E-01	.86394E+00	-.48279E-02	-.99388E-07
47	.13058E+02	-.66324E-03	.11113E-01	.86820E+00	-.46656E-02	.34210E-07
48	.13479E+02	-.63523E-03	.10958E-01	.87244E+00	-.45024E-02	.14133E-06
49	.13918E+02	-.60772E-03	.10805E-01	.87666E+00	-.43384E-02	.22259E-06
50	.14373E+02	-.58079E-03	.10653E-01	.88085E+00	-.41736E-02	.27833E-06
51	.14845E+02	-.55450E-03	.10499E-01	.88501E+00	-.40083E-02	.31294E-06
52	.15335E+02	-.52887E-03	.10341E-01	.88913E+00	-.38428E-02	.32160E-06
53	.15845E+02	-.50396E-03	.10177E-01	.89321E+00	-.36772E-02	.32365E-06
54	.16373E+02	-.47977E-03	.10006E-01	.89724E+00	-.35121E-02	.27080E-06
55	.16922E+02	-.45636E-03	.98265E-02	.90123E+00	-.33478E-02	.36100E-06
56	.17493E+02	-.43361E-03	.96372E-02	.90517E+00	-.31845E-02	-.86201E-07
57	.18085E+02	-.41195E-03	.94372E-02	.90906E+00	-.30233E-02	.13434E-05
58	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

59	.19537E+0[illegible]	-.37[illegible]E-03	[illegible]	[illegible]	[illegible]
60	.20000E+02	-.35521E-03	.87515E-02	.92040E+00	-.26818E-02	0.

J = 5 R(J) = .12474E+00

I	TRR	TTT	TZZ	TRT	TRZ	TZT	EPSL
1	.38579E-02	.38579E-02	.30324E-02	-.12459E-03	-.11839E-02	.35403E-03	.10491E-02
2	.38672E-02	.39056E-02	.30700E-02	-.10559E-03	-.12104E-02	.38140E-03	.10498E-02
3	.38907E-02	.39645E-02	.31146E-02	-.87051E-04	-.12380E-02	.40967E-03	.10387E-02
4	.39115E-02	.40264E-02	.31627E-02	-.68644E-04	-.12619E-02	.43798E-03	.10389E-02
5	.39250E-02	.40854E-02	.32124E-02	-.50359E-04	-.12803E-02	.46570E-03	.10396E-02
6	.39317E-02	.41375E-02	.32622E-02	-.32409E-04	-.12927E-02	.49236E-03	.10398E-02
7	.39329E-02	.41807E-02	.33111E-02	-.15103E-04	-.12995E-02	.51763E-03	.10389E-02
8	.39301E-02	.42139E-02	.33586E-02	.12462E-05	-.13014E-02	.54125E-03	.10367E-02
9	.39243E-02	.42371E-02	.34042E-02	.16375E-04	-.12988E-02	.56304E-03	.10330E-02
10	.39161E-02	.42507E-02	.34476E-02	.30095E-04	-.12926E-02	.58287E-03	.10278E-02
11	.39060E-02	.42552E-02	.34886E-02	.42288E-04	-.12833E-02	.60065E-03	.10212E-02
12	.38939E-02	.42515E-02	.35270E-02	.52910E-04	-.12715E-02	.61634E-03	.10132E-02
13	.38800E-02	.42405E-02	.35626E-02	.61968E-04	-.12577E-02	.62994E-03	.10038E-02
14	.38642E-02	.42229E-02	.35955E-02	.69513E-04	-.12424E-02	.64147E-03	.99325E-03
15	.38464E-02	.41997E-02	.36255E-02	.75625E-04	-.12258E-02	.65098E-03	.98149E-03
16	.38267E-02	.41715E-02	.36525E-02	.80401E-04	-.12083E-02	.65854E-03	.96863E-03
17	.38050E-02	.41392E-02	.36766E-02	.83948E-04	-.11902E-02	.66424E-03	.95476E-03
18	.37813E-02	.41032E-02	.36978E-02	.86380E-04	-.11716E-02	.66817E-03	.93996E-03
19	.37558E-02	.40641E-02	.37161E-02	.87806E-04	-.11527E-02	.67044E-03	.92432E-03
20	.37284E-02	.40225E-02	.37316E-02	.88334E-04	-.11336E-02	.67116E-03	.90791E-03
21	.36993E-02	.39788E-02	.37442E-02	.88065E-04	-.11144E-02	.67044E-03	.89083E-03
22	.36687E-02	.39334E-02	.37541E-02	.87097E-04	-.10952E-02	.66839E-03	.87315E-03
23	.36366E-02	.38867E-02	.37614E-02	.85517E-04	-.10761E-02	.66513E-03	.85495E-03
24	.36032E-02	.38389E-02	.37662E-02	.83409E-04	-.10570E-02	.66079E-03	.83633E-03
25	.35687E-02	.37903E-02	.37685E-02	.80849E-04	-.10380E-02	.65548E-03	.81734E-03
26	.35331E-02	.37412E-02	.37685E-02	.77907E-04	-.10191E-02	.64931E-03	.79408E-03
27	.34966E-02	.36918E-02	.37662E-02	.74647E-04	-.10002E-02	.64240E-03	.77860E-03
28	.34593E-02	.36422E-02	.37618E-02	.71131E-04	-.98139E-03	.63487E-03	.75898E-03
29	.34213E-02	.35926E-02	.37554E-02	.67414E-04	-.96258E-03	.62681E-03	.73927E-03
30	.33828E-02	.35430E-02	.37469E-02	.63550E-04	-.94374E-03	.61833E-03	.71953E-03
31	.33436E-02	.34935E-02	.37365E-02	.59587E-04	-.92483E-03	.60951E-03	.69980E-03
32	.33041E-02	.34442E-02	.37243E-02	.55576E-04	-.90581E-03	.60045E-03	.68013E-03
33	.32640E-02	.33951E-02	.37101E-02	.51562E-04	-.88662E-03	.59120E-03	.66056E-03
34	.32235E-02	.33461E-02	.36940E-02	.47590E-04	-.86723E-03	.58183E-03	.64112E-03
35	.31826E-02	.32973E-02	.36761E-02	.43702E-04	-.84760E-03	.57239E-03	.62183E-03
36	.31411E-02	.32486E-02	.36562E-02	.39940E-04	-.82769E-03	.56289E-03	.60271E-03
37	.30992E-02	.31998E-02	.36343E-02	.36341E-04	-.80748E-03	.55337E-03	.58378E-03
38	.30567E-02	.31510E-02	.36103E-02	.32937E-04	-.78696E-03	.54382E-03	.56505E-03
39	.30137E-02	.31020E-02	.35843E-02	.29758E-04	-.76613E-03	.53423E-03	.54654E-03
40	.29700E-02	.30527E-02	.35561E-02	.26827E-04	-.74500E-03	.52459E-03	.52823E-03
41	.29256E-02	.30030E-02	.35255E-02	.24161E-04	-.72359E-03	.51486E-03	.51015E-03
42	.28805E-02	.29528E-02	.34927E-02	.21770E-04	-.70195E-03	.50502E-03	.49229E-03
43	.28346E-02	.29021E-02	.34574E-02	.19657E-04	-.68011E-03	.49502E-03	.47465E-03
44	.27879E-02	.28507E-02	.34197E-02	.17820E-04	-.65814E-03	.48484E-03	.45725E-03
45	.27404E-02	.27985E-02	.33795E-02	.16249E-04	-.63610E-03	.47443E-03	.44007E-03
46	.26921E-02	.27456E-02	.33368E-02	.14930E-04	-.61404E-03	.46376E-03	.42312E-03
47	.26429E-02	.26918E-02	.32915E-02	.13846E-04	-.59203E-03	.45280E-03	.40641E-03
48	.25927E-02	.26371E-02	.32437E-02	.12975E-04	-.57014E-03	.44153E-03	.38994E-03
49	.25417E-02	.25815E-02	.31934E-02	.12295E-04	-.54842E-03	.42995E-03	.37371E-03
50	.24898E-02	.25250E-02	.31406E-02	.11781E-04	-.52693E-03	.41803E-03	.35774E-03
51	.24370E-02	.24676E-02	.30854E-02	.11411E-04	-.50572E-03	.40579E-03	.34204E-03
52	.23832E-02	.24093E-02	.30279E-02	.11161E-04	-.48484E-03	.39323E-03	.32661E-03
53	.23286E-02	.23502E-02	.29682E-02	.11011E-04	-.46433E-03	.38037E-03	.31147E-03
54	.22732E-02	.22903E-02	.29063E-02	.10940E-04	-.44421E-03	.36723E-03	.29662E-03
55	.22169E-02	.22296E-02	.28425E-02	.10931E-04	-.42453E-03	.35383E-03	.28209E-03
56	.21598E-02	.21683E-02	.27769E-02	.10968E-04	-.40529E-03	.34021E-03	.26789E-03
57	.21020E-02	.21065E-02	.27096E-02	.11038E-04	-.38653E-03	.32641E-03	.254[illegible]E-03

60 [illegible] .18653E-02 .27740E-[illegible] .9579[illegible]-04 -.6[illegible]12E-03 [illegible]55524E-03 .24117E-0[illegible]

J = 10 R(J) = .35411E+00

I	Z	U	V	W	P	DIV
1	.40000E+01	0.	.73883E-01	.87608E+00	-.10016E-01	-.91448E-05
2	.40748E+01	-.93629E-03	.73661E-01	.87606E+00	-.99365E-02	.36658E-03
3	.41525E+01	-.15588E-02	.73422E-01	.87606E+00	-.97633E-02	.15800E-03
4	.42331E+01	-.19688E-02	.73162E-01	.87610E+00	-.95443E-02	.35612E-04
5	.43168E+01	-.22393E-02	.72879E-01	.87622E+00	-.93190E-02	.53986E-05
6	.44036E+01	-.24171E-02	.72571E-01	.87641E+00	-.91041E-02	.67280E-06
7	.44940E+01	-.25329E-02	.72235E-01	.87668E+00	-.89046E-02	-.28047E-05
8	.45878E+01	-.26068E-02	.71870E-01	.87701E+00	-.87211E-02	-.58352E-05
9	.46851E+01	-.26515E-02	.71476E-01	.87740E+00	-.85528E-02	-.71064E-05
10	.47861E+01	-.26758E-02	.71052E-01	.87786E+00	-.83980E-02	-.71515E-05
11	.48911E+01	-.26855E-02	.70598E-01	.87837E+00	-.82547E-02	-.67328E-05
12	.50000E+01	-.26846E-02	.70112E-01	.87894E+00	-.81212E-02	-.61711E-05
13	.51131E+01	-.26761E-02	.69596E-01	.87956E+00	-.79958E-02	-.55617E-05
14	.52306E+01	-.26620E-02	.69048E-01	.88023E+00	-.78771E-02	-.49598E-05
15	.53525E+01	-.26437E-02	.68470E-01	.88095E+00	-.77637E-02	-.44060E-05
16	.54792E+01	-.26222E-02	.67861E-01	.88173E+00	-.76546E-02	-.39209E-05
17	.56107E+01	-.25985E-02	.67223E-01	.88256E+00	-.75487E-02	-.35096E-05
18	.57472E+01	-.25730E-02	.66557E-01	.88344E+00	-.74454E-02	-.31681E-05
19	.58890E+01	-.25462E-02	.65864E-01	.88437E+00	-.73438E-02	-.28861E-05
20	.60361E+01	-.25184E-02	.65146E-01	.88536E+00	-.72434E-02	-.26495E-05
21	.61890E+01	-.24899E-02	.64405E-01	.88640E+00	-.71436E-02	-.24431E-05
22	.63477E+01	-.24607E-02	.63642E-01	.88749E+00	-.70439E-02	-.22525E-05
23	.65125E+01	-.24310E-02	.62861E-01	.88864E+00	-.69441E-02	-.20657E-05
24	.66836E+01	-.24007E-02	.62065E-01	.88985E+00	-.68438E-02	-.18743E-05
25	.68612E+01	-.23699E-02	.61255E-01	.89110E+00	-.67426E-02	-.16738E-05
26	.70457E+01	-.23385E-02	.60436E-01	.89242E+00	-.66404E-02	-.14636E-05
27	.72372E+01	-.23064E-02	.59609E-01	.89378E+00	-.65369E-02	-.12462E-05
28	.74361E+01	-.22734E-02	.58778E-01	.89521E+00	-.64319E-02	-.10263E-05
29	.76426E+01	-.22395E-02	.57945E-01	.89668E+00	-.63254E-02	-.80981E-06
30	.78570E+01	-.22046E-02	.57113E-01	.89821E+00	-.62172E-02	-.60299E-06
31	.80796E+01	-.21680E-02	.56283E-01	.89979E+00	-.61071E-02	-.41156E-06
32	.83107E+01	-.21310E-02	.55457E-01	.90143E+00	-.59951E-02	-.24026E-06
33	.85508E+01	-.20921E-02	.54637E-01	.90311E+00	-.58812E-02	-.92470E-07
34	.88000E+01	-.20517E-02	.53823E-01	.90484E+00	-.57654E-02	.29803E-07
35	.90588E+01	-.20098E-02	.53016E-01	.90662E+00	-.56474E-02	.12588E-06
36	.93274E+01	-.19663E-02	.52215E-01	.90844E+00	-.55275E-02	.19624E-06
37	.96064E+01	-.19212E-02	.51420E-01	.91031E+00	-.54055E-02	.24224E-06
38	.98961E+01	-.18746E-02	.50629E-01	.91222E+00	-.52816E-02	.26591E-06
39	.10197E+02	-.18265E-02	.49841E-01	.91416E+00	-.51556E-02	.26969E-06
40	.10509E+02	-.17770E-02	.49054E-01	.91614E+00	-.50278E-02	.25621E-06
41	.10833E+02	-.17262E-02	.48266E-01	.91815E+00	-.48980E-02	.22817E-06
42	.11170E+02	-.16744E-02	.47475E-01	.92019E+00	-.47664E-02	.18816E-06
43	.11520E+02	-.16216E-02	.46678E-01	.92225E+00	-.46332E-02	.13861E-06
44	.11883E+02	-.15680E-02	.45871E-01	.92433E+00	-.44983E-02	.81726E-07
45	.12260E+02	-.15139E-02	.45054E-01	.92643E+00	-.43619E-02	.19469E-07
46	.12651E+02	-.14594E-02	.44223E-01	.92855E+00	-.42242E-02	-.46462E-07
47	.13058E+02	-.14048E-02	.43375E-01	.93067E+00	-.40854E-02	-.11458E-06
48	.13479E+02	-.13503E-02	.42509E-01	.93280E+00	-.39456E-02	-.18363E-06
49	.13918E+02	-.12960E-02	.41622E-01	.93493E+00	-.38051E-02	-.25252E-06
50	.14373E+02	-.12422E-02	.40713E-01	.93706E+00	-.36641E-02	-.32012E-06
51	.14845E+02	-.11890E-02	.39780E-01	.93918E+00	-.35228E-02	-.38631E-06
52	.15335E+02	-.11366E-02	.38822E-01	.94129E+00	-.33817E-02	-.44733E-06
53	.15845E+02	-.10852E-02	.37839E-01	.94339E+00	-.32408E-02	-.51275E-06
54	.16373E+02	-.10347E-02	.36831E-01	.94548E+00	-.31007E-02	-.54816E-06
55	.16922E+02	-.98551E-03	.35798E-01	.94754E+00	-.29615E-02	-.65463E-06
56	.17493E+02	-.93746E-03	.34739E-01	.94958E+00	-.28237E-02	-.54048E-06
57	.18065E+02	-.89100E-03	.33657E-01	.95160E+00	-.26873E-02	-.91265E-06
[illegible]	[illegible]	[illegible]	[illegible]	.95359E+00	-.25526E-02	-.50906E-06

59	[illegible]	[illegible]	[illegible]			
60	.20000E+02	-.75798E-03	.30249F-01	.95744F+00	-.23862F-02	0.

J = 10 R(J) = .35411E+00

I.	TRR,	TTT,	TZZ,	THT,	TRZ,	TZT,	EPSL
1	.53907F-02	.62314F-02	.43941F-02	.27642E-03	-.23019E-02	.19102F-02	.19042F-02
2	.52591F-02	.61052E-02	.44270F-02	.29241F-03	-.22803E-02	.18780F-02	.16586F-02
3	.51379F-02	.59871E-02	.44622F-02	.31105F-03	-.22622E-02	.18471E-02	.18157E-02
4	.50257E-02	.58755E-02	.44986F-02	.33164E-03	-.22462E-02	.18176F-02	.17752E-02
5	.49218E-02	.57691F-02	.45357E-02	.35362F-03	-.22313F-02	.17893E-02	.17366F-02
6	.48257F-02	.56668E-02	.45729E-02	.37642F-03	-.22170E-02	.17621F-02	.16995F-02
7	.47371E-02	.55678F-02	.46099F-02	.39957E-03	-.22031E-02	.17359E-02	.16638F-02
8	.46554E-02	.54713E-02	.46466E-02	.42261F-03	-.21892E-02	.17107E-02	.16291E-02
9	.45802E-02	.53770E-02	.46825E-02	.44511E-03	-.21753E-02	.16862E-02	.15954F-02
10	.45111E-02	.52843E-02	.47175E-02	.46670F-03	-.21612E-02	.16625F-02	.15625F-02
11	.44475E-02	.51928F-02	.47514F-02	.48703E-03	-.21467E-02	.16395E-02	.15303E-02
12	.43890E-02	.51025E-02	.47839F-02	.50578F-03	-.21319E-02	.16171F-02	.14986E-02
13	.43349E-02	.50131E-02	.48147E-02	.52267E-03	-.21166E-02	.15951E-02	.14673F-02
14	.42849E-02	.49245E-02	.48438E-02	.53747F-03	-.21009E-02	.15736F-02	.14364F-02
15	.42384E-02	.48367F-02	.48708E-02	.54997E-03	-.20848E-02	.15526E-02	.14058F-02
16	.41950F-02	.47496F-02	.48956F-02	.56001F-03	-.20682E-02	.15319F-02	.13754E-02
17	.41541E-02	.46634F-02	.49180F-02	.56748F-03	-.20511E-02	.15115F-02	.13452F-02
18	.41153F-02	.45780F-02	.49379E-02	.57229F-03	-.20336E-02	.14914F-02	.13151E-02
19	.40783E-02	.44936F-02	.49550E-02	.57441F-03	-.20157E-02	.14716F-02	.12852E-02
20	.40426F-02	.44103E-02	.49693F-02	.57385F-03	-.19975E-02	.14520F-02	.12553E-02
21	.40079E-02	.43282F-02	.49806E-02	.57065E-03	-.19788E-02	.14326F-02	.12255F-02
22	.39739E-02	.42475E-02	.49889E-02	.56491E-03	-.19598E-02	.14134F-02	.11958E-02
23	.39402F-02	.41682E-02	.49941E-02	.55673E-03	-.19405E-02	.13945E-02	.11662F-02
24	.39065F-02	.40905E-02	.49960F-02	.54626E-03	-.19208F-02	.13757E-02	.11366E-02
25	.38726E-02	.40145F-02	.49948E-02	.53370E-03	-.19007F-02	.13571F-02	.11072F-02
26	.38383E-02	.39404E-02	.49902F-02	.51924F-03	-.18803F-02	.13387F-02	.10779E-02
27	.38032E-02	.38681E-02	.49824E-02	.50311F-03	-.18593E-02	.13206E-02	.10487E-02
28	.37673E-02	.37977F-02	.49711F-02	.48556E-03	-.18378F-02	.13025F-02	.10197E-02
29	.37303E-02	.37292F-02	.49565F-02	.46686E-03	-.18158E-02	.12847E-02	.99082F-03
30	.36922F-02	.36626E-02	.49384E-02	.44727F-03	-.17930E-02	.12671E-02	.96217F-03
31	.36526E-02	.35979F-02	.49169E-02	.42707E-03	-.17694E-02	.12495F-02	.93373F-03
32	.36117F-02	.35349E-02	.48919E-02	.40655F-03	-.17450E-02	.12321F-02	.90551E-03
33	.35693F-02	.34735E-02	.48632F-02	.38596F-03	-.17196F-02	.12147F-02	.87752E-03
34	.35253F-02	.34135F-02	.48310E-02	.36558F-03	-.16931F-02	.11973F-02	.84977F-03
35	.34798E-02	.33549E-02	.47950E-02	.34564E-03	-.16656E-02	.11799E-02	.82225F-03
36	.34327F-02	.32973E-02	.47553E-02	.32636F-03	-.16368E-02	.11624E-02	.79497E-03
37	.33841E-02	.32404F-02	.47117F-02	.30793F-03	-.16068E-02	.11447E-02	.76793F-03
38	.33340E-02	.31842E-02	.46643E-02	.29052E-03	-.15755E-02	.11267E-02	.74112F-03
39	.32825E-02	.31282E-02	.46130E-02	.27425F-03	-.15429E-02	.11083F-02	.71456E-03
40	.32296E-02	.30722F-02	.45577F-02	.25921F-03	-.15090E-02	.10895E-02	.68824F-03
41	.31754E-02	.30160F-02	.44985F-02	.24546E-03	-.14739F-02	.10703E-02	.66217F-03
42	.31201E-02	.29594F-02	.44354F-02	.23302E-03	-.14377E-02	.10504E-02	.63637E-03
43	.30635E-02	.29021E-02	.43685F-02	.22187E-03	-.14003F-02	.10299F-02	.61084F-03
44	.30059E-02	.28440E-02	.42977F-02	.21197E-03	-.13618E-02	.10087E-02	.58560F-03
45	.29473E-02	.27850F-02	.42232F-02	.20326E-03	-.13225F-02	.98680F-03	.56068E-03
46	.28877F-02	.27248F-02	.41451E-02	.19565E-03	-.12824E-02	.96409E-03	.53610E-03
47	.28272E-02	.26636F-02	.40637F-02	.18903E-03	-.12417F-02	.94059E-03	.51190F-03
48	.27659F-02	.26013E-02	.39790E-02	.18330E-03	-.12004F-02	.91629E-03	.48811E-03
49	.27039F-02	.25379E-02	.38913E-02	.17833F-03	-.11588E-02	.89121E-03	.46475F-03
50	.26411F-02	.24735E-02	.38009E-02	.17401F-03	-.11171F-02	.86537F-03	.44186F-03
51	.25775F-02	.24081F-02	.37080E-02	.17021F-03	-.10752E-02	.83882E-03	.41948F-03
52	.25134E-02	.23419F-02	.36128E-02	.16683E-03	-.10335F-02	.81161E-03	.39765F-03
53	.24486F-02	.22751E-02	.35157E-02	.16375E-03	-.99194F-03	.78380E-03	.37638F-03
54	.23835F-02	.22078F-02	.34170E-02	.16090E-03	-.95079F-03	.75547F-03	.35572E-03
55	.23176F-02	.21401F-02	.33169E-02	.15819E-03	-.91011F-03	.72672F-03	.33569F-03
56	.22513E-02	.20723E-02	.32158E-02	.15554E-03	-.87002F-03	.69764F-03	.31631F-03
57	.21848E-02	.20046F-02	.31140E-02	[illegible]			

59	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
60	.17611E-02	.12916E-02	.21622E-02	.446[illegible]E-03	-.[illegible]392E-03	.30622E-03

59	[illegible]	
60	.20027E-03	

J = 15 R(J) = .70524E+00

I	Z	U	V	W	P	DIV
1	.40000E+01	0.	.61689E-01	.10814E+01	-.11122E-02	-.27718E-04
2	.40708E+01	.28672E-04	.61555E-01	.10814E+01	-.14195E-02	-.67620E-04
3	.41525E+01	-.29078E-04	.61424E-01	.10813E+01	-.17039E-02	.89094E-04
4	.42331E+01	-.11666E-03	.61295E-01	.10812E+01	-.19460E-02	.16774E-03
5	.43168E+01	-.20638E-03	.61168E-01	.10809E+01	-.21370E-02	.16572E-03
6	.44038E+01	-.28954E-03	.61041E-01	.10806E+01	-.22811E-02	.13741E-03
7	.44940E+01	-.36400E-03	.60916E-01	.10803E+01	-.23881E-02	.11062E-03
8	.45878E+01	-.42905E-03	.60790E-01	.10799E+01	-.24671E-02	.90106E-04
9	.46851E+01	-.48460E-03	.60665E-01	.10795E+01	-.25249E-02	.73994E-04
10	.47861E+01	-.53127E-03	.60539E-01	.10790E+01	-.25668E-02	.60712E-04
11	.48911E+01	-.57018E-03	.60412E-01	.10785E+01	-.25963E-02	.49592E-04
12	.50000E+01	-.60261E-03	.60283E-01	.10780E+01	-.26165E-02	.40275E-04
13	.51131E+01	-.62975E-03	.60150E-01	.10774E+01	-.26293E-02	.32484E-04
14	.52306E+01	-.65266E-03	.60014E-01	.10768E+01	-.26365E-02	.26008E-04
15	.53525E+01	-.67220E-03	.59872E-01	.10761E+01	-.26393E-02	.20669E-04
16	.54792E+01	-.68910E-03	.59723E-01	.10754E+01	-.26386E-02	.16309E-04
17	.56107E+01	-.70393E-03	.59566E-01	.10747E+01	-.26351E-02	.12780E-04
18	.57472E+01	-.71712E-03	.59399E-01	.10739E+01	-.26294E-02	.99499E-05
19	.58890E+01	-.72901E-03	.59219E-01	.10731E+01	-.26219E-02	.77044E-05
20	.60361E+01	-.73985E-03	.59025E-01	.10723E+01	-.26129E-02	.59410E-05
21	.61890E+01	-.74980E-03	.58815E-01	.10714E+01	-.26026E-02	.45708E-05
22	.63477E+01	-.75897E-03	.58586E-01	.10704E+01	-.25912E-02	.35173E-05
23	.65125E+01	-.76742E-03	.58337E-01	.10695E+01	-.25787E-02	.27162E-05
24	.66836E+01	-.77517E-03	.58065E-01	.10684E+01	-.25653E-02	.21146E-05
25	.68612E+01	-.78218E-03	.57768E-01	.10674E+01	-.25510E-02	.16693E-05
26	.70457E+01	-.78843E-03	.57444E-01	.10663E+01	-.25357E-02	.13452E-05
27	.72372E+01	-.79386E-03	.57092E-01	.10651E+01	-.25195E-02	.11143E-05
28	.74361E+01	-.79838E-03	.56709E-01	.10640E+01	-.25023E-02	.95431E-06
29	.76426E+01	-.80191E-03	.56294E-01	.10627E+01	-.24840E-02	.84731E-06
30	.78570E+01	-.80438E-03	.55845E-01	.10615E+01	-.24647E-02	.77914E-06
31	.80796E+01	-.80571E-03	.55362E-01	.10602E+01	-.24442E-02	.73854E-06
32	.83107E+01	-.80581E-03	.54842E-01	.10589E+01	-.24226E-02	.71665E-06
33	.85508E+01	-.80463E-03	.54286E-01	.10576E+01	-.23998E-02	.70660E-06
34	.88000E+01	-.80210E-03	.53692E-01	.10562E+01	-.23758E-02	.70312E-06
35	.90588E+01	-.79819E-03	.53061E-01	.10548E+01	-.23505E-02	.70238E-06
36	.93274E+01	-.79288E-03	.52392E-01	.10534E+01	-.23239E-02	.70166E-06
37	.96060E+01	-.78614E-03	.51686E-01	.10520E+01	-.22960E-02	.69924E-06
38	.98961E+01	-.77801E-03	.50942E-01	.10505E+01	-.22668E-02	.69420E-06
39	.10197E+02	-.76848E-03	.50162E-01	.10490E+01	-.22363E-02	.68621E-06
40	.10509E+02	-.75762E-03	.49346E-01	.10476E+01	-.22045E-02	.67545E-06
41	.10833E+02	-.74547E-03	.48495E-01	.10461E+01	-.21715E-02	.66240E-06
42	.11170E+02	-.73210E-03	.47610E-01	.10446E+01	-.21372E-02	.64774E-06
43	.11520E+02	-.71759E-03	.46693E-01	.10432E+01	-.21016E-02	.63226E-06
44	.11883E+02	-.70204E-03	.45744E-01	.10417E+01	-.20649E-02	.61673E-06
45	.12260E+02	-.68554E-03	.44766E-01	.10402E+01	-.20270E-02	.60183E-06
46	.12651E+02	-.66819E-03	.43761E-01	.10388E+01	-.19880E-02	.58814E-06
47	.13058E+02	-.65011E-03	.42728E-01	.10373E+01	-.19479E-02	.57606E-06
48	.13479E+02	-.63140E-03	.41672E-01	.10359E+01	-.19067E-02	.56578E-06
49	.13918E+02	-.61219E-03	.40592E-01	.10345E+01	-.18645E-02	.55740E-06
50	.14373E+02	-.59257E-03	.39492E-01	.10331E+01	-.18213E-02	.55055E-06
51	.14845E+02	-.57266E-03	.38372E-01	.10317E+01	-.17771E-02	.54558E-06
52	.15335E+02	-.55256E-03	.37235E-01	.10304E+01	-.17321E-02	.54004E-06
53	.15845E+02	-.53237E-03	.36082E-01	.10290E+01	-.16862E-02	.53936E-06
54	.16373E+02	-.51217E-03	.34916E-01	.10277E+01	-.16395E-02	.52634E-06
55	.16922E+02	-.49206E-03	.33738E-01	.10265E+01	-.15920E-02	.54300E-06
56	.17493E+02	-.47212E-03	.32551E-01	.10253E+01	-.15439E-02	.50683E-06
57	.18085E+02	-.45227E-03	.31355E-01	.10241E+01	-.14951E-02	.44753E-06

60	.20000E+0[illegible]	-.368[illegible]5E-03	.27743E-01	.10206E+01	-.13694E-02	0.	

J = 15 R(J) = .70524E+00

I	TRR	TTT	TZZ	TRT	TRZ	TZT	EPSL
1	.15360E-02	.16090E-02	.11016E-02	-.11297E-03	.64822E-04	-.21220E-03	.25951E-03
2	.15951E-02	.16085E-02	.11051E-02	-.64863E-04	.55924E-04	-.21083E-03	.26177E-03
3	.16665E-02	.16093E-02	.11105E-02	-.16989E-04	.46860E-04	-.20995E-03	.26510E-03
4	.17465E-02	.16116E-02	.11178E-02	.30118E-04	.37420E-04	-.20938E-03	.26938E-03
5	.18316E-02	.16151E-02	.11271E-02	.76036E-04	.27453E-04	-.20896E-03	.27450E-03
6	.19187E-02	.16199E-02	.11383E-02	.12045E-03	.16854E-04	-.20854E-03	.28030E-03
7	.20050E-02	.16257E-02	.11514E-02	.16313E-03	.55523E-05	-.20798E-03	.28665E-03
8	.20884E-02	.16325E-02	.11665E-02	.20395E-03	-.64960E-05	-.20716E-03	.29337E-03
9	.21672E-02	.16399E-02	.11834E-02	.24282E-03	-.19314E-04	-.20597E-03	.30034E-03
10	.22402E-02	.16478E-02	.12021E-02	.27975E-03	-.32906E-04	-.20434E-03	.30740E-03
11	.23068E-02	.16560E-02	.12226E-02	.31477E-03	-.47264E-04	-.20220E-03	.31445E-03
12	.23666E-02	.16643E-02	.12449E-02	.34796E-03	-.62367E-04	-.19950E-03	.32136E-03
13	.24196E-02	.16725E-02	.12688E-02	.37939E-03	-.78180E-04	-.19620E-03	.32806E-03
14	.24659E-02	.16804E-02	.12943E-02	.40918E-03	-.94662E-04	-.19229E-03	.33447E-03
15	.25061E-02	.16879E-02	.13214E-02	.43742E-03	-.11176E-03	-.18774E-03	.34055E-03
16	.25404E-02	.16949E-02	.13501E-02	.46418E-03	-.12940E-03	-.18256E-03	.34625E-03
17	.25694E-02	.17013E-02	.13802E-02	.48955E-03	-.14752E-03	-.17676E-03	.35154E-03
18	.25935E-02	.17069E-02	.14117E-02	.51356E-03	-.16603E-03	-.17033E-03	.35642E-03
19	.26133E-02	.17118E-02	.14445E-02	.53625E-03	-.18483E-03	-.16331E-03	.36087E-03
20	.26291E-02	.17158E-02	.14785E-02	.55765E-03	-.20384E-03	-.15569E-03	.36489E-03
21	.26414E-02	.17190E-02	.15137E-02	.57775E-03	-.22294E-03	-.14750E-03	.36847E-03
22	.26505E-02	.17213E-02	.15499E-02	.59657E-03	-.24204E-03	-.13876E-03	.37163E-03
23	.26566E-02	.17228E-02	.15869E-02	.61408E-03	-.26101E-03	-.12948E-03	.37435E-03
24	.26599E-02	.17233E-02	.16246E-02	.63028E-03	-.27975E-03	-.11970E-03	.37665E-03
25	.26608E-02	.17231E-02	.16629E-02	.64516E-03	-.29816E-03	-.10942E-03	.37853E-03
26	.26593E-02	.17220E-02	.17015E-02	.65871E-03	-.31612E-03	-.98695E-04	.37997E-03
27	.26555E-02	.17201E-02	.17403E-02	.67092E-03	-.33352E-03	-.87538E-04	.38098E-03
28	.26496E-02	.17175E-02	.17790E-02	.68180E-03	-.35028E-03	-.75987E-04	.38156E-03
29	.26417E-02	.17142E-02	.18174E-02	.69134E-03	-.36629E-03	-.64089E-04	.38169E-03
30	.26318E-02	.17101E-02	.18553E-02	.69958E-03	-.38147E-03	-.51860E-04	.38136E-03
31	.26199E-02	.17053E-02	.18924E-02	.70653E-03	-.39572E-03	-.39373E-04	.38057E-03
32	.26061E-02	.16999E-02	.19285E-02	.71221E-03	-.40898E-03	-.26669E-04	.37930E-03
33	.25905E-02	.16937E-02	.19633E-02	.71666E-03	-.42117E-03	-.13804E-04	.37754E-03
34	.25731E-02	.16869E-02	.19965E-02	.71992E-03	-.43224E-03	-.83515E-06	.37529E-03
35	.25538E-02	.16795E-02	.20280E-02	.72204E-03	-.44213E-03	.12175E-04	.37253E-03
36	.25329E-02	.16713E-02	.20575E-02	.72306E-03	-.45080E-03	.25162E-04	.36927E-03
37	.25102E-02	.16624E-02	.20847E-02	.72302E-03	-.45822E-03	.38059E-04	.36548E-03
38	.24858E-02	.16527E-02	.21094E-02	.72198E-03	-.46436E-03	.50798E-04	.36118E-03
39	.24597E-02	.16422E-02	.21315E-02	.71998E-03	-.46921E-03	.63311E-04	.35636E-03
40	.24321E-02	.16309E-02	.21507E-02	.71706E-03	-.47277E-03	.75529E-04	.35103E-03
41	.24029E-02	.16187E-02	.21668E-02	.71327E-03	-.47503E-03	.87385E-04	.34519E-03
42	.23722E-02	.16055E-02	.21796E-02	.70864E-03	-.47601E-03	.98814E-04	.33887E-03
43	.23400E-02	.15914E-02	.21895E-02	.70321E-03	-.47573E-03	.10975E-03	.33206E-03
44	.23064E-02	.15762E-02	.21958E-02	.69699E-03	-.47423E-03	.12015E-03	.32480E-03
45	.22714E-02	.15598E-02	.21985E-02	.69003E-03	-.47154E-03	.12994E-03	.31711E-03
46	.22351E-02	.15424E-02	.21977E-02	.68232E-03	-.46770E-03	.13908E-03	.30901E-03
47	.21975E-02	.15237E-02	.21933E-02	.67389E-03	-.46278E-03	.14753E-03	.30053E-03
48	.21587E-02	.15038E-02	.21853E-02	.66475E-03	-.45682E-03	.15524E-03	.29171E-03
49	.21187E-02	.14826E-02	.21736E-02	.65490E-03	-.44990E-03	.16219E-03	.28259E-03
50	.20775E-02	.14602E-02	.21584E-02	.64436E-03	-.44208E-03	.16836E-03	.27317E-03
51	.20352E-02	.14364E-02	.21396E-02	.63312E-03	-.43344E-03	.17371E-03	.26354E-03
52	.19919E-02	.14114E-02	.21173E-02	.62120E-03	-.42406E-03	.17826E-03	.25371E-03
53	.19475E-02	.13852E-02	.20918E-02	.60861E-03	-.41401E-03	.18197E-03	.24374E-03
54	.19021E-02	.13577E-02	.20629E-02	.59536E-03	-.40339E-03	.18487E-03	.23365E-03
55	.18559E-02	.13289E-02	.20310E-02	.58147E-03	-.39227E-03	.18696E-03	.22351E-03
56	.18088E-02	.12991E-02	.19961E-02	.56697E-03	-.38074E-03	.18825E-03	.21335E-03
57	.17609E-02	.12682E-02	.19585E-02	.55188E-03	-.36888E-03	.18876E-03	.20320E-03

60	.70916E-03	.39575E-03	.47317E-03	.18793E-03	.12885E-03	-.19072E-04	.46486E-04

J = 20 R(J) = .12428E+01

I	Z	U	V	W	P	DIV
1	.40000E+01	0.	-.52000E-02	.10000E+01	-.33071E-03	.55204E-05
2	.40748E+01	-.10704E-04	-.51745E-02	.10000E+01	-.32829E-03	.23238E-04
3	.41525E+01	-.11181E-04	-.51475E-02	.10000E+01	-.32747E-03	.87025E-05
4	.42331E+01	-.95199E-05	-.51185E-02	.10000E+01	-.32898E-03	-.71436E-05
5	.43168E+01	-.96849E-05	-.50873E-02	.10001E+01	-.33372E-03	-.15248E-04
6	.44038E+01	-.13594E-04	-.50535E-02	.10001E+01	-.34146E-03	-.17164E-04
7	.44940E+01	-.21287E-04	-.50167E-02	.10001E+01	-.35111E-03	-.15553E-04
8	.45878E+01	-.31766E-04	-.49767E-02	.10002E+01	-.36152E-03	-.12648E-04
9	.46851E+01	-.43773E-04	-.49330E-02	.10002E+01	-.37180E-03	-.99732E-05
10	.47861E+01	-.56265E-04	-.48853E-02	.10002E+01	-.38142E-03	-.80630E-05
11	.48911E+01	-.68522E-04	-.48335E-02	.10003E+01	-.39017E-03	-.67266E-05
12	.50000E+01	-.80067E-04	-.47772E-02	.10003E+01	-.39804E-03	-.56212E-05
13	.51131E+01	-.90568E-04	-.47162E-02	.10004E+01	-.40507E-03	-.45850E-05
14	.52306E+01	-.99801E-04	-.46503E-02	.10004E+01	-.41136E-03	-.36205E-05
15	.53525E+01	-.10764E-03	-.45793E-02	.10005E+01	-.41701E-03	-.27640E-05
16	.54792E+01	-.11405E-03	-.45031E-02	.10005E+01	-.42211E-03	-.20224E-05
17	.56107E+01	-.11905E-03	-.44216E-02	.10006E+01	-.42673E-03	-.13832E-05
18	.57472E+01	-.12271E-03	-.43345E-02	.10006E+01	-.43093E-03	-.83241E-06
19	.58890E+01	-.12512E-03	-.42419E-02	.10007E+01	-.43473E-03	-.34208E-06
20	.60361E+01	-.12638E-03	-.41435E-02	.10008E+01	-.43815E-03	.33585E-07
21	.61890E+01	-.12659E-03	-.40393E-02	.10009E+01	-.44118E-03	.36085E-06
22	.63477E+01	-.12585E-03	-.39293E-02	.10010E+01	-.44381E-03	.62651E-06
23	.65125E+01	-.12426E-03	-.38133E-02	.10011E+01	-.44601E-03	.83740E-06
24	.66836E+01	-.12189E-03	-.36913E-02	.10012E+01	-.44776E-03	.10001E-05
25	.68612E+01	-.11882E-03	-.35632E-02	.10013E+01	-.44905E-03	.11210E-05
26	.70457E+01	-.11513E-03	-.34290E-02	.10014E+01	-.44987E-03	.12058E-05
27	.72372E+01	-.11087E-03	-.32886E-02	.10015E+01	-.45018E-03	.12600E-05
28	.74361E+01	-.10608E-03	-.31419E-02	.10017E+01	-.45001E-03	.12885E-05
29	.76426E+01	-.10083E-03	-.29890E-02	.10018E+01	-.44934E-03	.12955E-05
30	.78570E+01	-.95141E-04	-.28296E-02	.10020E+01	-.44818E-03	.12848E-05
31	.80796E+01	-.89062E-04	-.26639E-02	.10021E+01	-.44656E-03	.12596E-05
32	.83107E+01	-.82627E-04	-.24916E-02	.10023E+01	-.44449E-03	.12227E-05
33	.85508E+01	-.75870E-04	-.23127E-02	.10025E+01	-.44200E-03	.11764E-05
34	.88000E+01	-.68827E-04	-.21271E-02	.10026E+01	-.43913E-03	.11226E-05
35	.90588E+01	-.61534E-04	-.19347E-02	.10028E+01	-.43590E-03	.10629E-05
36	.93274E+01	-.54031E-04	-.17355E-02	.10030E+01	-.43237E-03	.99874E-06
37	.96064E+01	-.46359E-04	-.15291E-02	.10032E+01	-.42858E-03	.93111E-06
38	.98961E+01	-.38561E-04	-.13157E-02	.10034E+01	-.42455E-03	.86099E-06
39	.10197E+02	-.30685E-04	-.10949E-02	.10036E+01	-.42035E-03	.78917E-06
40	.10509E+02	-.22779E-04	-.86666E-03	.10039E+01	-.41600E-03	.71639E-06
41	.10833E+02	-.14897E-04	-.63088E-03	.10041E+01	-.41155E-03	.64328E-06
42	.11170E+02	-.70925E-05	-.38738E-03	.10043E+01	-.40703E-03	.57046E-06
43	.11520E+02	.57951E-06	-.13605E-03	.10045E+01	-.40249E-03	.49854E-06
44	.11883E+02	.80631E-05	.12321E-03	.10048E+01	-.39795E-03	.42809E-06
45	.12260E+02	.15303E-04	.39048E-03	.10050E+01	-.39345E-03	.35971E-06
46	.12651E+02	.22244E-04	.66579E-03	.10052E+01	-.38900E-03	.29400E-06
47	.13058E+02	.28836E-04	.94915E-03	.10055E+01	-.38462E-03	.23153E-06
48	.13479E+02	.35030E-04	.12405E-02	.10057E+01	-.38034E-03	.17288E-06
49	.13918E+02	.40782E-04	.15396E-02	.10059E+01	-.37616E-03	.11859E-06
50	.14373E+02	.46053E-04	.18463E-02	.10062E+01	-.37209E-03	.69188E-07
51	.14845E+02	.50809E-04	.21603E-02	.10064E+01	-.36814E-03	.25071E-07
52	.15335E+02	.55025E-04	.24810E-02	.10066E+01	-.36429E-03	-.13404E-07
53	.15845E+02	.58681E-04	.28081E-02	.10068E+01	-.36055E-03	-.45497E-07
54	.16373E+02	.61769E-04	.31406E-02	.10070E+01	-.35689E-03	-.73519E-07
55	.16922E+02	.64284E-04	.34779E-02	.10072E+01	-.35329E-03	-.86473E-07
56	.17493E+02	.66252E-04	.38190E-02	.10073E+01	-.34973E-03	-.13454E-06
57	.18085E+02	.67705E-04	.4162?E-02	[illegible]	[illegible]	[illegible]
58	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

60	[illegible]	[illegible]	.51985E-02	.10079E+01	-.34279E-03	0.

J = 20 R(J) = .12429E+01

I	TRR	TTT	TZZ	TRT	TRZ	TZT	EPSL
1	.24165E-03	.10967E-03	.12341E-03	-.11398E-03	.66633E-05	-.88449E-04	.10593E-04
2	.27075E-03	.11074E-03	.12539E-03	-.11270E-03	.65620E-05	-.88952E-04	.11036E-04
3	.28160E-03	.11191E-03	.12754E-03	-.11136E-03	.65325E-05	-.89538E-04	.11553E-04
4	.29425E-03	.11317E-03	.12987E-03	-.10993E-03	.65975E-05	-.90207E-04	.12150E-04
5	.30871E-03	.11452E-03	.13238E-03	-.10840E-03	.67801E-05	-.90957E-04	.12830E-04
6	.32491E-03	.11597E-03	.13507E-03	-.10676E-03	.71038E-05	-.91780E-04	.13594E-04
7	.34277E-03	.11751E-03	.13794E-03	-.10498E-03	.75909E-05	-.92670E-04	.14442E-04
8	.36213E-03	.11915E-03	.14098E-03	-.10307E-03	.82626E-05	-.93614E-04	.15372E-04
9	.38280E-03	.12088E-03	.14420E-03	-.10101E-03	.91376E-05	-.94603E-04	.16381E-04
10	.40456E-03	.12270E-03	.14758E-03	-.98803E-04	.10232E-04	-.95621E-04	.17463E-04
11	.42715E-03	.12461E-03	.15112E-03	-.96434E-04	.11558E-04	-.96653E-04	.18610E-04
12	.45031E-03	.12661E-03	.15481E-03	-.93908E-04	.13125E-04	-.97685E-04	.19816E-04
13	.47375E-03	.12868E-03	.15863E-03	-.91228E-04	.14938E-04	-.98699E-04	.21070E-04
14	.49719E-03	.13083E-03	.16257E-03	-.88401E-04	.16998E-04	-.99679E-04	.22361E-04
15	.52034E-03	.13305E-03	.16662E-03	-.85434E-04	.19302E-04	-.10061E-03	.23679E-04
16	.54293E-03	.13533E-03	.17076E-03	-.82339E-04	.21842E-04	-.10146E-03	.25011E-04
17	.56470E-03	.13766E-03	.17498E-03	-.79129E-04	.24609E-04	-.10224E-03	.26345E-04
18	.58542E-03	.14004E-03	.17926E-03	-.75819E-04	.27589E-04	-.10290E-03	.27668E-04
19	.60488E-03	.14245E-03	.18358E-03	-.72424E-04	.30766E-04	-.10346E-03	.28968E-04
20	.62289E-03	.14490E-03	.18793E-03	-.68960E-04	.34120E-04	-.10388E-03	.30232E-04
21	.63932E-03	.14736E-03	.19227E-03	-.65444E-04	.37631E-04	-.10415E-03	.31451E-04
22	.65403E-03	.14983E-03	.19660E-03	-.61892E-04	.41277E-04	-.10427E-03	.32612E-04
23	.66695E-03	.15230E-03	.20090E-03	-.58319E-04	.45036E-04	-.10422E-03	.33705E-04
24	.67801E-03	.15476E-03	.20515E-03	-.54740E-04	.48883E-04	-.10399E-03	.34722E-04
25	.68721E-03	.15719E-03	.20932E-03	-.51168E-04	.52796E-04	-.10359E-03	.35656E-04
26	.69453E-03	.15960E-03	.21341E-03	-.47614E-04	.56751E-04	-.10300E-03	.36500E-04
27	.70002E-03	.16197E-03	.21740E-03	-.44088E-04	.60727E-04	-.10222E-03	.37248E-04
28	.70373E-03	.16429E-03	.22127E-03	-.40597E-04	.64703E-04	-.10125E-03	.37898E-04
29	.70573E-03	.16656E-03	.22502E-03	-.37147E-04	.68659E-04	-.10009E-03	.38446E-04
30	.70611E-03	.16876E-03	.22863E-03	-.33741E-04	.72576E-04	-.98741E-04	.38892E-04
31	.70498E-03	.17090E-03	.23208E-03	-.30382E-04	.76438E-04	-.97211E-04	.39236E-04
32	.70246E-03	.17296E-03	.23539E-03	-.27068E-04	.80228E-04	-.95503E-04	.39479E-04
33	.69866E-03	.17495E-03	.23853E-03	-.23798E-04	.83934E-04	-.93623E-04	.39623E-04
34	.69373E-03	.17685E-03	.24150E-03	-.20567E-04	.87541E-04	-.91580E-04	.39672E-04
35	.68777E-03	.17867E-03	.24430E-03	-.17370E-04	.91040E-04	-.89383E-04	.39630E-04
36	.68094E-03	.18041E-03	.24693E-03	-.14200E-04	.94418E-04	-.87041E-04	.39502E-04
37	.67335E-03	.18207E-03	.24938E-03	-.11050E-04	.97667E-04	-.84564E-04	.39293E-04
38	.66513E-03	.18364E-03	.25166E-03	-.79103E-05	.10078E-03	-.81965E-04	.39008E-04
39	.65640E-03	.18513E-03	.25377E-03	-.47723E-05	.10374E-03	-.79255E-04	.38655E-04
40	.64728E-03	.18654E-03	.25571E-03	-.16264E-05	.10655E-03	-.76447E-04	.38238E-04
41	.63787E-03	.18787E-03	.25749E-03	.15370E-05	.10919E-03	-.73554E-04	.37765E-04
42	.62826E-03	.18912E-03	.25909E-03	.47268E-05	.11166E-03	-.70589E-04	.37242E-04
43	.61857E-03	.19030E-03	.26053E-03	.79514E-05	.11395E-03	-.67566E-04	.36675E-04
44	.60885E-03	.19141E-03	.26181E-03	.11218E-04	.11604E-03	-.64498E-04	.36069E-04
45	.59919E-03	.19245E-03	.26292E-03	.14533E-04	.11793E-03	-.61402E-04	.35430E-04
46	.58966E-03	.19342E-03	.26388E-03	.17901E-04	.11960E-03	-.58292E-04	.34763E-04
47	.58030E-03	.19431E-03	.26467E-03	.21322E-04	.12103E-03	-.55182E-04	.34074E-04
48	.57116E-03	.19513E-03	.26529E-03	.24798E-04	.12222E-03	-.52089E-04	.33365E-04
49	.56228E-03	.19588E-03	.26576E-03	.28323E-04	.12314E-03	-.49028E-04	.32641E-04
50	.55369E-03	.19654E-03	.26605E-03	.31893E-04	.12379E-03	-.46016E-04	.31905E-04
51	.54540E-03	.19711E-03	.26618E-03	.35497E-04	.12413E-03	-.43068E-04	.31160E-04
52	.53742E-03	.19757E-03	.26613E-03	.39122E-04	.12416E-03	-.40200E-04	.30408E-04
53	.52975E-03	.19792E-03	.26590E-03	.42750E-04	.12386E-03	-.37428E-04	.29652E-04
54	.52238E-03	.19813E-03	.26550E-03	.46361E-04	.12321E-03	-.34765E-04	.28891E-04
55	.51529E-03	.19820E-03	.26492E-03	.49930E-04	.12220E-03	-.32225E-04	.28129E-04
56	.50845E-03	.19810E-03	.26416E-03	.53429E-04	.12083E-03	-.29819E-04	.27365E-04
57	.50184E-03	.19781E-03	.26322E-03	.56829E-04	.11908E-03	-.27557E-04	.26601E-04

AO .48298E-03 .19564E-[illegible] .2597[illegible]E-03 .[illegible]112E-04 .11162E-03 -.21691E-[illegible] .2430[illegible]E-04

MASS FLUX CALCULATION
WEST FLUX = -.800403E+01 EAST FLUX = .800588E+01 NORTH FLUX = -.194100E-02
NET OUT-FLUX = -.915790E-04
COMPARE PRESSURE GRADIENT AND REYNOLDS STRESS TERMS IN THE AXIAL MOMENTUM EQUATION

J = 1

T,DZP,TURAN,DZZZ,DRZR

2	.1703E-01	-.2093E-01	-.4274E-04	-.2089E-01
3	.1130E-01	-.2106E-01	.4809E-04	-.2110E-01
4	.7273E-02	-.2112E-01	.1407E-03	-.2126E-01
5	.4726E-02	-.2118E-01	.2255E-03	-.2140E-01
6	.3175E-02	-.2123E-01	.2959E-03	-.2152E-01
7	.2270E-02	-.2126E-01	.3489E-03	-.2160E-01
8	.1768E-02	-.2125E-01	.3842E-03	-.2163E-01
9	.1507E-02	-.2120E-01	.4036E-03	-.2160E-01
10	.1386E-02	-.2110E-01	.4096E-03	-.2151E-01
11	.1342E-02	-.2097E-01	.4047E-03	-.2138E-01
12	.1341E-02	-.2081E-01	.3916E-03	-.2120E-01
13	.1358E-02	-.2062E-01	.3726E-03	-.2099E-01
14	.1380E-02	-.2041E-01	.3495E-03	-.2076E-01
15	.1401E-02	-.2018E-01	.3239E-03	-.2051E-01
16	.1415E-02	-.1994E-01	.2969E-03	-.2024E-01
17	.1420E-02	-.1970E-01	.2694E-03	-.1997E-01
18	.1417E-02	-.1944E-01	.2422E-03	-.1968E-01
19	.1404E-02	-.1918E-01	.2157E-03	-.1939E-01
20	.1383E-02	-.1891E-01	.1902E-03	-.1910E-01
21	.1355E-02	-.1863E-01	.1662E-03	-.1880E-01
22	.1319E-02	-.1835E-01	.1436E-03	-.1849E-01
23	.1279E-02	-.1807E-01	.1227E-03	-.1819E-01
24	.1233E-02	-.1778E-01	.1034E-03	-.1788E-01
25	.1185E-02	-.1749E-01	.8578E-04	-.1757E-01
26	.1133E-02	-.1719E-01	.6973E-04	-.1726E-01
27	.1080E-02	-.1689E-01	.5518E-04	-.1695E-01
28	.1027E-02	-.1659E-01	.4203E-04	-.1663E-01
29	.9730E-03	-.1629E-01	.3016E-04	-.1632E-01
30	.9199E-03	-.1598E-01	.1942E-04	-.1600E-01
31	.8680E-03	-.1566E-01	.9687E-05	-.1567E-01
32	.8180E-03	-.1535E-01	.8108E-06	-.1535E-01
33	.7703E-03	-.1502E-01	-.7338E-05	-.1502E-01
34	.7252E-03	-.1470E-01	-.1488E-04	-.1468E-01
35	.6830E-03	-.1436E-01	-.2193E-04	-.1434E-01
36	.6439E-03	-.1402E-01	-.2857E-04	-.1400E-01
37	.6081E-03	-.1368E-01	-.3488E-04	-.1364E-01
38	.5758E-03	-.1333E-01	-.4091E-04	-.1329E-01
39	.5459E-03	-.1297E-01	-.4669E-04	-.1293E-01
40	.5198E-03	-.1261E-01	-.5223E-04	-.1256E-01
41	.4957E-03	-.1225E-01	-.5755E-04	-.1219E-01
42	.4745E-03	-.1188E-01	-.6261E-04	-.1182E-01
43	.4556E-03	-.1151E-01	-.6741E-04	-.1144E-01
44	.4386E-03	-.1113E-01	-.7193E-04	-.1106E-01
45	.4232E-03	-.1076E-01	-.7613E-04	-.1068E-01
46	.4090E-03	-.1039E-01	-.7999E-04	-.1031E-01
47	.3959E-03	-.1001E-01	-.8350E-04	-.9931E-02
48	.3833E-03	-.9645E-02	-.8662E-04	-.9558E-02
49	.3713E-03	-.9279E-02	-.8934E-04	-.9190E-02
50	.3594E-03	-.8917E-02	-.9167E-04	-.8826E-02
51	.3475E-03	-.8561E-02	-.9358E-04	-.8467E-02
52	.3356E-03	-.8210E-02	-.9508E-04	-.8115E-02
53	.3234E-03	-.7866E-02	-.9618E-04	-.7770E-02

REFERENCES

Chieng, C. C., Jakubowski, A. K. and Schetz, J. A. (1974) "Investigation of the Turbulent Properties of the Wake Behind Self-Propelled, Axisymmetric Bodies," Virginia Polytechnic Institute and State University Report No. VPI-Aero-025.

Grabowski, W. J., et al. (1976) "ICWAKE COMPUTER CODE -- Mathematical Analysis and Finite-Difference Formulation," Flow Research Report No. 69, in preparation.

Gran, R. L. (1976) private communication.

Hanjalic, K. and Launder, B. E. (1972) "A Reynolds Stress Model of Turbulence and Its Application to Thin Shear Flows," J. Fl. Mech. 52, part 4, 609-638.

Launder, B. E., Reece, G. J. and Rodi, W. (1975) "Progress in the Development of a Reynolds-Stress Turbulence Closure," J. Fl. Mech. 68, part 3, 537-566.

Mager, A. (1972) "Dissipation and Breakdown of a Wing-Tip Vortex," J. Fl. Mech. 55, part 4, 609-628.

Roache, P. J. (1972) Computational Fluid Dynamics, Hermosa Publishers.

Schwartz, L. W. and Bernstein, S. (1975) "Documentation of a Propeller Analysis Computer Program," Flow Research Note No. 67 (APL/JHU POR-3686).

Swanson, Jr., R. C., Schetz, J. A. and Jakubowski, A. K. (1974) "Turbulent Wake Behind Slender Bodies Including Self-Propelled Configurations," Virginia Polytechnic Institute and State University Report No. VPI-Aero-024.

Appendix

The following pages contain a listing of the ICWAKE computer code. The routines appear in the following order:

ICWAKE
BLAYER
PRØPWV
PRØPU
PTURB
ELINT1
ELINT2
QSF
CØEFF
NEWT
UPCØND
DATIN
IUNI
NØZERØ
STRESS2
PRESSR
DIVCØN
FLXCHK
TERMCHK
BLKTRI
BLKTR1
PRØD
PRØDP
CPRØD
CPRØDP
PPADD
PSGF
BSRH
PPSGF
PPSPF
CØMPB
TQLRAT

ICWAKE

```
      PROGRAM ICWAKE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE41,
     X TAPE42,TAPE43)
      REAL K,KS,KSI,KI,KM,K2I
      REAL NPRZE,NPRZW,NUZTE,NUZTW,NUZZE,NUZZW,NLRRN,NURRS,NURTN,NURTS,
     1  NURZN,NURZS,NURTC
      COMMON/VLIF/RP(41),JND(41)
      COMMON/TRICAT/DU(61),DV(61),DW(61),UT(61),VT(61),WT(61)
      COMMON/TRICAT2/AU(61),AV(61),AW(61),BU(61),BV(61),BW(61),CU(61),
     1  CV(61),CW(61),EU(61),EV(61),EW(61),FE(32),FE2(32)
      COMMON/VEL/U(61,41),V(61,41),W(61,41),P(61,41),DIV(61,41)
      COMMON/DAT/X(61),Y(61),R(61),Z(61),F(61),S(61),S2(61),C1(61)
     1,C2(61),C3(61),C4(61),C5(61),C6(61),C7(61),C10(61),C11(61),C12(61)
     2, TA,RE,H,K,N,M,AM,MM,MI2,MSI,KSI,HH,KH,E(61),F2(61),FH(61),XH(61)
     3, AX,BX,AY,BY,NUMBER,NC,MC,EPS,NSTRT,GEF(61),TH(61),ITURB,NTURB
      COMMON/PRESCO/ AM(61),AN(61),BM(61),BN(61),CM(61),CN(61),EM(61),
     1  EMS(61),FM(61),FMS(61),RH(61),SH(61),SHS(61),THNN,WORK(600),
     2  BC01,BC02,BC03,SI,TI,SIS,SIREI,MM2,IFL6,+2I,K2I
      COMMON/UPSTRM/WH(61),WHRR(61),WI,VI,ALPH,RMAX,ZMAX,REI,THETAI,RR
      COMMON/VISC/ARTVIS(61)
      COMMON/STRESS/TRR(61,32),TZZ(61,32),TTT(61,32),TRZ(61,32),
     1  TRT(61,32),TZT(61,32),EPSL(61,32)
      COMMON/2/MPOINT,RT(32),UL(32),VL(32),WL(32),UU(32),VV(32),WW(32),
     1 WU(32),WV(32),UV(32),EPSIL(32),ISCHETZ,IDEP,IBLPRP
      COMMON /BLP/REBLF,BRD,BLN,BUIN,RPROP,ALAMB,NB,NBEL,BELR(20),
     X           BELC(20),BELANG(20),BELTH(20),BELDEL(20),DWDUM(32)
      NAMELIST/DAT1/NUMBER,NSTRT,NTMX,NPRNT,M1,M2,M3,M11,M22,M33,TA
     1 ,NTCHK,CONCRIT
      NAMELIST /DAT2/N,M,NC,MC,AX,AY,ZMAX,RMAX,EPS,YMAX,ZCC,RCC,ZINITL,
     X           NTURB
      NAMELIST /CAT3/RE,WI,VI,ALPH,IMAGER,ITURB,IDECOUP,BRD,BLN,BUIN,
     X       RPRCP,ALAMB,NB,NBEL,BELR,BELC,BELANG,BELTH,BELDEL
      NAMELIST /CAT4/MPOINT,RT,UL,VL,WL,UU,VV,WW,WU,WV,UV,EPSIL,
     X              ISCHETZ,IDEP,IBLPRP
      NAMELIST/DAT5/ISWEEPX,ISWEEPY
      NAMELIST/DAT6/ARTVIS
      DATA ND,NUMBER,NSTRT/61,1,0/
      DATA ICON,NTCHK,NPRNT,NTMX,IPAGE,ISWEEPX,ISWEEPY/0,10,30,30,1,1,1
      DATA IDECOUP,CONCRIT,ZINITL,ZCC,RCC,EPS,YMAX,XMAX/0,1.E-9,0.,1.,
     1 1.,1.E-6,0.5,1.0/
      DATA M1,M2,M3,M11,M22,M33/1,1,1,1,8,1/
      DATA N,M,NC,MC,MTURB/40,16,10,8,0/
      DATA TA/0.1/
      DATA ZMAX,RMAX,AX,AY/20.,6.,3.076,5.8/
      DATA IMAGER/0/
      DATA ISCHETZ,IDEP,IBLPRP/0,0,0/
      READ(5,DAT1)
      TAI2 = 2./TA
C     .....OBTAIN DATE AND HOUR OF RUNNING
      CALL HOUR(ITIME)
      CALL DATE(IDAY)
      PRINT 781,IDAY,ITIME
      PRINT 87
      PRINT 783,NUMBER
      PRINT 92,NSTRT
      PRINT 791,NTMX,NPRNT
      PRINT 792,M1,M2,M3
```

TCWAKE

```
      IF(NUMBER.EQ.1.AND.NSTRT.EQ.0)GO TO 60
      READ(41) U,V,W,P,DIV,N,M,NC,MC,AX,BX,AY,BY,ZMAX,RMAX,EPS,YMAX,RE,
     1  WI,VI,ALPH,IMAGER,ITURB,TRR,TTT,TZZ,TRT,TRZ,TZT,EPSL
     1  ,R,Z,RCC,ZCC,ZINITL,MTURB,ISCHETZ,IDEP,IBLPRP
      IF(NSTRT.NE.0)GO TO 61
      GO TO 62
60    CONTINUE
      READ(5,DAT2)
      IF(MTURB.EQ.0) MTURB=M
   62 READ(5,DAT3)
      IF(IMAGER.EQ.1) GO TO 61
      READ(5,DAT4)
      RMAX = RT(MPOINT)
      IF(IBLPRP.EQ.0) GO TO 61
C
C  DO BOUNDARY LAYER AND PROPELLER CALCULATION
      REBLP=RE
      CALL BLAYER
      CALL PROPWV
      CALL PROPU
      CALL PTURB
C
   61 CONTINUE
      READ(5,DAT5)
      READ(5,DAT6)
C     .....SET PARAMETERS
      X(N) = XMAX
      Y(M) = YMAX
      NM = N-1
      MM = M-1
      NM2 = N-2
      MM2 = M-2
      H = 1./NM
      K = Y(M)/MM
      DO 404J = 1,M
404   JND(J) = (J-1)*NC
      PRINT 793,N,M
      PRINT 795,H,K
      PRINT 786,ZMAX,RMAX
      PRINT 787,X(N),Y(M)
C     .....COMPUTE COEFFICIENTS USED IN CALCULATIONS
   63 CALL COEFF(ZINITL,RCC,ZCC)
      DO 57J = 2,M
      FE(J) = F(J)*E(J)
   57 FE2(J) = 2.*FE(J)
      PRINT 785,NC,MC,ZCC,RCC,ZINITL,AX,BX,AY,BY,EPS
      PRINT 799
      DO 101I = 1,N
      PRINT 83,I,X(I),Z(I),XH(I),ZH(I)
101   CONTINUE
      PRINT 800
      DO 102J = 1,M
      PRINT 83,J,Y(J),R(J),RH(J)
102   CONTINUE
      IF(ISWEEPX.EQ.0.AND.ISWEEPY.EQ.1) PRINT 780
      IF(ISWEEPX.EQ.1.AND.ISWEEPY.EQ.0) PRINT 779
```

ICWAKE

```
      IF(ISWEEPX.EQ.0.AND.ISWEEPY.EQ.0) GO TO 776
      PRINT 87
      WRITE(6,DAT3)
      WRITE(6,DAT5)
      WRITE(6,DAT6)
      PRINT 782,RE
      PRINT 794,TA
      PRINT 7844
      PRINT 7845
      PRINT 7846
      PRINT 7847
      PRINT 7848
C     .....COMPUTE INITIAL CONDITIONS
      CALL UPCOND(IMAGER,IDECOUP)
C     ..............................................................
C     .....BEGIN TIME MARCHING WITH INITIAL DATA OR DATA FROM PREVIOUS
C     .....  CALCULATION.
C     ..............................................................
      N1 = NSTRT+1
      N2 = NSTRT+NTMX
C     .....TIME STEP LOOP
      DO 666NT = N1,N2
  324 CONTINUE
C     .....OPTION TO SKIP VERTICAL SWEEP
      IF(ISWEEPY.EQ.0) GO TO 4001
C     ..............................................................
C     .....VERTICAL SWEEP - IMPLICIT IN X
C     ..............................................................
C     FIRST ROW
      DO 1009I = 2,N
      UT(I) = 0.
      VT(I) = 0.
 1009 WT(I) = W(I)
C     .....SELECT ROW.  J INDEX REFERS TO ROWS.
      DO 501J = 2,MM
      JC=JND(J)
      JS=JND(J-1)
      JN=JND(J+1)
C     .....SET UP TRIDI MATRIX
      DO 68I = 2,NM
      LC=I+JC
      WABS=ABS(W(LC))
      A = C1(I)-S(I)*(W(LC)+ARTVIS(I)*WABS)
      B = C2(I)+S2(I)*ARTVIS(I)*WABS
      C = C3(I)+S(I)*(W(LC)-ARTVIS(I)*WABS)
      AU(I) = A
      AV(I) = A
      AW(I) = A
      BU(I) = B
      BV(I) = B
      BW(I) = B
      CU(I) = C
      CV(I) = C
      CW(I) = C
   68 CONTINUE
C     .....SET UP NONHOMOGENOUS TERMS FOR TRIDI INVERSION
```

ICWAKE

```
      DO 37I = 2,NM
      LC = I+JC
      LS = I+JS
      LN = I+JN
      DZP = S(I)*(P(LC)-P(LC-1)+P(LS)-P(LS-1))
      DRP=F(J)*(P(LC)-P(LS)+P(LC-1)-P(LS-1))
      FU=F(J)*U(LC)
      T1=C4(J)+FU
      T2=C7(J)
      T2P=C5(J)
      T3=C6(J)-FU
  510 DU(I)=U(LS)*T1+U(LC)*T2+U(LN)*T3+V(LC)*V(LC)/R(J)-DRP
      DV(I)=V(LS)*T1+V(LC)*(T2-U(LC)/R(J))+V(LN)*T3
   37 DW(I) = W(LS)*T1+W(LC)*T2P+W(LN)*T3-DZP
      IF(ITURB.EQ.0) GO TO 632
      IF(IDECOUP.EQ.1) GO TO 632
      DO 633 I=2,NM
      LC = I+JC
      LN = I+JN
      LS = I+JS
      TURBU = F(J)*(TRR(LN)-TRR(LS))+(TRR(LC)-TTT(LC))*E(J)
      TURBV = F(J)*(TRT(LN)-TRT(LS))+2.*TRT(LC)*E(J)
      TURBW = F(J)*(TRZ(LN)-TRZ(LS))+TRZ(LC)*E(J)
      TURBU = TURBU+S(I)*(TRZ(LC+1)-TRZ(LC-1))
      TURBV = TURBV+S(I)*(TZT(LC+1)-TZT(LC-1))
      TURBW = TURBW+S(I)*(TZZ(LC+1)-TZZ(LC-1))
      DU(I) = DU(I)-TURBU
      DV(I) = DV(I)-TURBV
  633 DW(I) = DW(I)-TURBW
  632 CONTINUE
C     .....MOVE RESULTS IN TEMPORARY STORAGE TO PERMANENT LOCATION
      DO 38I = 2,N
      L = I+JS
      U(L) = UT(I)
      V(L) = VT(I)
   38 W(L) = WT(I)
C     .....INVERT TRIDI MATRIX FOR V AND W
      DU(1) = U(1,J)
      CV(1) = V(1,J)
      DW(1) = W(1,J)
      EU(1) = 0.
      EV(1) = 0.
      EW(1) = 0.
      DO 50I = 2,NM
      IM = I-1
      ZU = 1./(BU(I)+AU(I)*EU(IM))
      ZV = 1./(BV(I)+AV(I)*EV(IM))
      ZW = 1./(BW(I)+AW(I)*EW(IM))
      EU(I) = -CU(I)*ZU
      EV(I) = -CV(I)*ZV
      EW(I) = -CW(I)*ZW
      DU(I) = (DU(I)-AU(I)*DU(IM))*ZU
      CV(I) = (DV(I)-AV(I)*DV(IM))*ZV
C     .....APPLY PARABOLIC OUTFLOW (EAST) BOUNDARY CONDITIONS
   50 DW(I) = (DW(I)-AW(I)*DW(IM))*ZW
      AA = 2.*S(N)*W(N,J)
```

ICWAKE

```
      EVM = 1./(TAI2+AA*(1.-EV(NM)))
      EWM = 1./(TAI2+AA*(1.-EW(NM)))
      RHSW = -F(J)*U(N,J)*(W(N,J+1)-W(N,J-1))
      RHSW = RHSW-2.*S(N)*(P(N,J)+P(N,J-1)-P(N-1,J)-P(N-1,J-1))
      RHSW = RHSW+C6(J)*W(N,J+1)+C4(J)*W(N,J-1)+C5(J)*W(N,J)
      RHSV = -E(J)*U(N,J)*V(N,J)-F(J)*U(N,J)*(V(N,J+1)-V(N,J-1))
      RHSV = RHSV+C6(J)*V(N,J+1)+C4(J)*V(N,J-1)+C7(J)*V(N,J)
      IF(ITURB.EQ.0) GC TO 640
      IF(IDECOUP.EQ.1) GO TO 640
      LN = N+JND(J+1)
      LC = N+JND(J)
      LS = N+JND(J-1)
      TURBV = F(J)*(TRT(LN)-TRT(LS))+2.*TRT(LC)*E(J)
      TURBW = F(J)*(TRZ(LN)-TRZ(LS))+TRZ(LC)*E(J)
      RHSW = RHSW-TURBW
      RHSV = RHSV-TURBV
  640 CONTINUE
      VT(N) = (AA*DV(NM)+RHSV)*EVM
      WT(N) = (AA*DW(NM)+RHSW)*EWM
      DO 51I = 1,NM2
      L = N-I
      LP = L+1
      VT(L) = DV(L)+EV(L)*VT(LP)
   51 WT(L) = DW(L)+EW(L)*WT(LP)
      IF(J.NE.2) GO TO 537
C     .....SATISFY B.C. ON AXIS
      DO 3221I = 2,N
 3221 W(I) = 4.*W(I,2)/3.-W(I,3)/3.
  537 CONTINUE
C     .....COMPUTE U AT EAST BOUNDARY
      TEM1 = U(N,J-1)*R(J-1)*E(J)
      DWXNJM = 3.*W(N,J-1)-4.*W(NM,J-1)+W(NM2,J-1)
      DWXNJ = 3.*WT(N)-4.*WT(NM)+WT(NM2)
      UT(N) = TEM1-0.25*RH(J-1)*S(N)*(DWXNJM+DWXNJ)/(FH(J-1)*R(J))
C     .....INVERT TRIDI MATRIX FOR U
      DO 475I = 1,NM2
      L = N-I
      LP = L+1
  475 UT(L) = DU(L)+EU(L)*UT(LP)
C     .....TRIDI INVERTED
501   CONTINUE
C     .....MOVE FINAL RCW OF U,V,W TO PERMANENT LCCATIONS
      JMN = JND(PM)
      DO 39I = 2,N
      L = I+JMN
      U(L) = UT(I)
      V(L) = VT(I)
39    W(L) = WT(I)
C     .....SATISFY CONDITION ON U AT N BOUNDARY
C     .....APPLY A BOUNCARY CONDITION TC SECOND ORDER
      DO 530I = 2,NM
  530 U(I,M) = (4.*U(I,MM)*R(MM)-U(I,MM2)*R(MM2))/(3.*R(M))
      TEM1 = -U(NM,M)+F(MM)*(U(N,MM)+U(NM,MM))/R(M)
      TEM2 = W(N,M)+W(N,MM)-W(NM,M)-W(NM,MM)
      U(N,M) = TEM1-SH(NM)*RH(MM)*TEM2/(R(M)*FH(MM))
C     .....COMPUTE DILATATION
```

```
ICWAKE
            DO 471I = 1,MM
            DO 471J = 1,MM
            DWZ = SH(I)*(W(I+1,J+1)+W(I+1,J)-W(I,J+1)-W(I,J))
            DURR = FM(J)*((U(I,J+1)+U(I+1,J+1))*R(J+1)-(U(I,J)+U(I+1,J))*R(J))
        471 DIV(I,J) = DWZ+DURR/RH(J)
            IF(ITURB.EQ.0) GO TO 540
      C     .....COMPUTE PEYNOLDS STRESSES
            CALL STRESS2(IDECOUP)
        540 CONTINUE
      C     .....COMPUTE PRESSURE AT EAST BOUNDARY
            P(N,MM) = 0.
            DO 476J = 2,MM
            L = M-J
            TEM1 = 0.5/F(L+1)*V(N,L+1)**2/R(L+1)
            IF(ITURB.EQ.0) GO TO 476
            IF(IDECOUP.EQ.1) GO TO 476
            TEM1 = TEM1+0.5/F(L+1)*(TTT(N,L+1)-TRR(N,L+1))/R(L+1)
            TEM1 = TEM1-0.5*(TRR(N,L+2)-TRR(N,L))
        476 P(N,L) = P(N,L+1)-TEM1
      C     .....COMPUTE PRESSURE IN INTERIOR
            CALL PRESSR(IDECCUP)
      C     ................................................................
      C     .....END OF VERTICAL SWEEP
      C     ................................................................
      C     .....OPTION TO SKIP HORIZONTAL SWEEP
       4001 IF(ISWEEPX.EQ.0) GO TO 4002
      C     .....HORIZONTAL SWEEP - IMPLICIT IN Y
      C     ................................................................
      C        PLACE FIRST COLUMN IN TEMPORARY LOCATIONS
            DO 40J = 2,M
            L = I+JND(J)
            UT(J) = U(L)
            VT(J) = V(L)
      40    WT(J) = W(L)
            WT(1) = W(1)
      C     .....SELECT COLUMN.  I INDEX REFERS TO COLUMNS.
            DO 601I = 2,N
      C     .....SET UP TRIDI MATRIX
            DO 42J = 2,MM
            LC = I+JND(J)
            LN = I+JND(J+1)
            LS = I+JND(J-1)
            FU = F(J)*U(LC)
            A = -(C4(J)+FU)
            B = C12(J)
            C = FU-C6(J)
            AU(J) = A
            AV(J) = A
            AW(J) = A
            BU(J) = B
            BV(J) = B
            BW(J) = C11(J)
            CU(J) = C
            CV(J) = C
            CW(J) = C
      42    CONTINUE
```

```
ICHANE
C     .....SET UP NONHOMOGENEOUS TERM FOR TRIDI INVERSION
      IF(I.EQ.N) GO TO 486
      DO 41J = 2,MM
      JC = JND(J)
      JS = JND(J-1)
      LC = I+JC
      LS = I+JS
      DZP=S(I)*(P(LC)-P(LC-1)+P(LS)-P(LS-1))
      DRP = F(J)*(P(LC)-P(LS)+P(LC-1)-P(LS-1))
      WABS=ABS(W(LC))
      T1=-C1(I)+S(I)*(W(LC)+ARTVIS(I)*WABS)
      T2=C10(I)-2.*S(I)*ARTVIS(I)*WABS
      T3 = -C3(I)-S(I)*(W(LC)-ARTVIS(I)*WABS)
      DU(J)=U(LC-1)*T1+U(LC)*T2+U(LC+1)*T3+V(LC)*V(LC)/R(J)-DRP
      DV(J)=V(LC-1)*T1+V(LC)*T2+V(LC+1)*T3-V(LC)*U(LC)/R(J)
   41 DW(J) = W(LC-1)*T1+W(LC)*T2+W(LC+1)*T3-DZP
      IF(ITURB.EQ.0) GO TO 634
      IF(IDECOUP.EQ.1) GO TO 634
      DO 635 J=2,MM
      JC = JND(J)
      JS = JND(J-1)
      JN=JND(J+1)
      LC = I+JC
      LS = I+JS
      LN = I+JN
      TURBU = S(I)*(TRZ(LC+1)-TRZ(LC-1))-TTT(LC)*E(J)
      TURBV = S(I)*(TZT(LC+1)-TZT(LC-1))
      TURBW = S(I)*(TZZ(LC+1)-TZZ(LC-1))
      TURBU = TURBU+F(J)*(TRR(LN)-TRR(LS))+TRR(LC)*E(J)
      TURBV = TURBV+F(J)*(TRT(LN)-TRT(LS))+2.*TRT(LC)*E(J)
      TURBW = TURBW+F(J)*(TRZ(LN)-TRZ(LS))+TRZ(LC)*E(J)
      DU(J) = DU(J)-TURBU
      DV(J) = DV(J)-TURBV
  635 DW(J) = DW(J)-TURBW
  634 CONTINUE
C     .....MOVE RESULTS IN TEMPORARY STORAGE TO PERMANENT LOCATION
      DO 43J = 2,M
      L = I-1+JND(J)
      U(L) = UT(J)
      V(L) = VT(J)
   43 W(L) = WT(J)
      W(I-1,1) = WT(1)
C     .....INVERT TRIDI FOR U AND V
      EU(1) = 0.
      EV(1) = 0.
      DU(1) = 0.
      DV(1) = 0.
      DUMM = DU(MM)
      DO 53J = 2,MM
      JM = J-1
      ZU = 1./(BU(J)+AU(J)*EU(JM))
      ZV = 1./(BV(J)+AV(J)*EV(JM))
      EU(J) = -CU(J)*ZU
      EV(J) = -CV(J)*ZV
      DU(J) = (DU(J)-AU(J)*DU(JM))*ZU
   53 DV(J) = (DV(J)-AV(J)*DV(JM))*ZV
```

ICWAKE

```
      VT(M) = V(I,M)
C     .....APPLY N BOUNDARY CONDITION TO SECOND ORDER
      APUM = -4.*R(MM)*AU(MM)*E(MM2)-BU(MM)
      BPUM = 3.*R(M)*AU(MM)*E(MM2)-CU(MM)
      DPUM = -DUMM
      TEM1 = DPUM/APUM
      TEM2 = BPUM/APUM
      UT(M) = (TEM1-DU(MM))/(EU(MM)+TEM2)
      DO 54J = 1,MM2
      L = M-J
      LP = L+1
      UT(L) = DU(L)+EU(L)*UT(LP)
   54 VT(L) = DV(L)+EV(L)*VT(LP)
C     .....TRIDI FOR U AND V INVERTED
C     .....INVERT TRIDI FOR W
C     .....APPLY NEUMANN CONDITION TO SECOND ORDER
      BIPP = 3.*CW(2)-AW(2)
      CIPP = -4.*CW(2)-BW(2)
      DIPP = -DW(2)
      EW(1) = -CIPP/BIPP
      DW(1) = DIPP/BIPP
      DO 55J = 2,MM
      JM = J-1
      ZW = 1./(BW(J)+AW(J)*EW(JM))
      EW(J) = -CW(J)*ZW
   55 DW(J) = (DW(J)-AW(J)*DW(JM))*ZW
      DO 56J = 1,MM
      L = M-J
   56 WT(L) = DW(L)+EW(L)*WT(L+1)
C     .....TRIDI FOR W INVERTED
      GO TO 601
  486 CONTINUE
C     .....INVERSION FOR FINAL COLUMN AT EAST BOUNDARY
C     .....ONLY EQUATIONS FOR V AND W INVERTED
      DO 477J = 2,MM
      LC = I+JND(J)
      LN = I+JND(J+1)
      LS = I+JND(J-1)
      TEM1 = 2.*V(N,J)/TA-E(J)*U(N,J)*V(N,J)
      TEM2 = 2.*S(N)*(V(N,J)-V(N-1,J))*W(N,J)
      DV(J) = TEM1-TEM2
      DW(J)= 2.*W(N,J)/TA-2.*S(N)*(W(N,J)-W(N-1,J))*W(N,J)
      IF(ITURB.EQ.0) GO TO 642
      IF(IDECOUP.EQ.1) GO TO 642
      TURBV = 0.
      TURBW = 0.
      TURBV = TURBV+F(J)*(TRT(LN)-TRT(LS))+2.*TRT(LC)*E(J)
      TURBW = TURBW+F(J)*(TRZ(LN)-TRZ(LS))+TRZ(LC)*E(J)
      DV(J) = DV(J)-TURBV
      DW(J) = DW(J)-TURBW
  642 CONTINUE
      DZP =  2.*S(N)*(P(N,J)+P(N,J-1)-P(N-1,J)-P(N-1,J-1))
  477 DW(J) = DW(J)-DZP
      DO 478J = 2,M
      U(I-1,J) = UT(J)
      V(I-1,J) = VT(J)
```

```
[illegible] W(I-1,J) = WT(J)
    W(I-1,1) = WT(1)
    EV(1) = 0.
    DV(1) = 0.
    VT(1) = 0.
    DO 479J = 2,MM
    ZV = 1./(BV(J)+AV(J)*EV(J-1))
    EV(J) = -CV(J)*ZV
479 DV(J) = (DV(J)-AV(J)*DV(J-1))*ZV
    VT(M) = V(1,M)
C     .....INVERT TRIDI FOR V
    DO 480J = 1,MM2
    L = M-J
480 VT(L) = DV(L)+EV(L)*VT(L+1)
C     .....APPLY NEUMANN CONDITION TO SECOND ORDER
    BIPP = 3.*CW(2)-AW(2)
    CIPP = -4.*CW(2)-BW(2)
    DIPP = -DW(2)
    EW(1) = -CIPP/BIPP
    DW(1) = DIPP/BIPP
    WT(M) = W(1,M)
    DO 481J = 2,MM
    ZW = 1./(BW(J)+AW(J)*EW(J-1))
    EW(J) = -CW(J)*ZW
481 DW(J) = (DW(J)-AW(J)*DW(J-1))*ZW
C     .....INVERT TRIDI FOR W
    DO 482J = 1,MM
    L = M-J
482 WT(L) = DW(L)+EW(L)*WT(L+1)
601 CONTINUE
C     .....MOVE FINAL COLUMNS OF V AND W TO PERMANENT LOCATION
    DO 484J = 2,M
    V(N,J) = VT(J)
484 W(N,J) = WT(J)
    W(N,1) = WT(1)
C     .....COMPUTE U AT EAST BOUNDARY
    U(N,1) = 0.
    DO 483J = 2,M
    TEM1 = U(N,J-1)*R(J-1)*E(J)
    DWXNJM = 3.*W(N,J-1)-4.*W(NM,J-1)+W(NM2,J-1)
    DWXNJ = 3.*W(N,J)-4.*W(NM,J)+W(NM2,J)
483 U(N,J) = TEM1-0.25*RW(J-1)*S(N)*(DWXNJM+DWXNJ)/(FH(J-1)*R(J))
    P(N,MM) = 0.
C     .....COMPUTE DILATATION
    DO 470I = 1,NM
    DO 470J = 1,MM
    DWZ = SH(I)*(W(I+1,J+1)+W(I+1,J)-W(I,J+1)-W(I,J))
    DURR = FH(J)*((U(I,J+1)+U(I+1,J+1))*R(J+1)-(U(I,J)+U(I+1,J))*R(J))
470 DIV(I,J) = DWZ+DURR/RH(J)
    IF(ITURB.EQ.0) GO TO 541
C     .....COMPUTE REYNOLDS STRESSES
    CALL STRESS2(IDECOUP)
541 CONTINUE
C     .....COMPUTE PRESSURE AT EAST BOUNDARY
    DO 485J = 2,MM
    L = M-J
```

```
ICWAKE
          TEM1 = 0.5/F(L+1)*V(N,L+1)**2/R(L+1)
          IF(ITURB.EQ.0) GC TO 485
          IF(IDECOUP.EQ.1) GO TO 485
          TEM1 = TEM1+0.5/F(L+1)*(TTT(N,L+1)-TRR(N,L+1))/R(L+1)
          TEM1 = TEM1-0.5*(TRR(N,L+2)-TRR(N,L))
    485 P(N,L) = P(N,L+1)-TEM1
C     .....COMPUTE PRESSURE IN INTERIOR
          CALL PRESSR(IDECCUP)
C     ..............................................................
C     .....END OF HORIZONTAL SWEEP
C     ..............................................................
C     .....END OF TIME STEP
   4002 CONTINUE
          IF(MOD(NT,NTCHK).NE.0) GO TO 4010
          IF(IPAGE.EQ.1) PRINT 8787
          IPAGE = 0
          CALL DIVCON (DIV,TA,NT,NM,MM,CONCRIT,ICON)
          IF(ICON.NE.1) GO TO 4010
          PRINT 8789,NT
          GO TO 4004
   4010 CONTINUE
          IF(NT.EQ.N2) GO TO 4011
          IF(MOD(NT,NPRNT).NE.0)GO TO666
   4004 PRINT 88,NT
          MK1 = M1
          MK2 = M2
          MK3 = M3
          IF(ICON.EQ.1) GO TO 4011
          IF(NT.NE.N2) GO TO 487
   4011 CONTINUE
          MK1 = M11
          MK2 = M22
          MK3 = M33
    487 CONTINUE
C   ... CHECKPOINT - SAVE DATA ON TAPE42
          REWIND 42
          WRITE(42) U,V,W,P,DIV,N,M,NC,MC,AX,BX,AY,BY,ZMAX,RMAX,EPS,YMAX,
       X            RE,WI,VI,ALPH,IMAGER,ITURB,TRR,TTT,TZZ,TRT,TR",TZT,EPSL,
       X            R,Z,RCC,ZCC,ZINITL,MTURB,ISCHEME,IDEP,IMLPRP
          DO 48J = MK1,MK2,MK3
          JMN = JND(J)
          L = 1+JMN
          I = 1
          PRINT 86,J,R(J),I,Z(I),U(L),V(L),W(L),P(L),DIV(L)
          DO 49I = 2,N
          L = I+JMN
          PRINT 83,I,Z(I),U(L),V(L),W(L),P(L),DIV(L)
  49      CONTINUE
          IF(ITURB.EQ.0) GO TO 48
          PRINT 84,J,R(J)
          DO 543I=1,N
          L = I+JMN
          PRINT 85,I,TRR(L),TTT(L),TZZ(L),TRT(L),TRZ(L),TZT(L),EPSL(L)
    543 CONTINUE
  48      CONTINUE
C     .....CHECK NET MASS FLUX
```

ICWAKE

```
      CALL FLXCHK(U,W,R,S,F,TA,N,M,H,K,FLUXW,FLUXE,FLUXN)
      TOTFLX = FLUXN+FLUXE+FLUXW
      PRINT 7001,FLUXW,FLUXE,FLUXN,TOTFLX
      IPAGE = 1
      IF(ICON.EQ.1) GO TO 6666
666   CONTINUE
 6666 CONTINUE
C     .....SAVE DATA ON TAPE42
      REWIND 42
      WRITE(42)U,V,W,P,DIV,N,M,NC,MC,AX,BX,AY,BY,ZMAX,RMAX,EPS,YMAX,RE,
     1  WI,VI,ALPH,IMAGER,ITURB,TRR,TTT,TZZ,TRT,TRZ,TZT,EPSL
     1  ,R,Z,RCC,ZCC,ZINITL,MTURB,ISCHETZ,IDEP,IOLPRP
C     ......TERM COMPARISON
      CALL TERMCHK
78    FORMAT(5F13.0)
79    FORMAT(7I5)
   83 FORMAT(1X,I5,6(6X,E12.5))
   84 FORMAT(1H1,*J =*,I3,5X,*R(J) =*,E12.5/5X,*I, TRR, TTT, TZZ, TRT, T
     1RZ, TZT, EPSL*)
   85 FORMAT(1X,I5,7(4X,E12.5))
   86 FORMAT(1H1,*J =*,I3,5X,*R(J) =*,E12.5,/5X,*I*,11X,*Z*,17X,*U*,17X,
     1*V*,17X,*W*,17X,*P*,16X,*DIV*,/(1X,I5,6(6X,E12.5)))
   87 FORMAT(1H1)
   88 FORMAT(1H1,20(/),25X,*ITERATION NUMBER*,2X,I5)
92    FORMAT(1H0,*ITERATION COUNT AT START, NSTRT =*,I6/)
      GO TO 775
  776 PRINT 777
  775 CONTINUE
  777 FORMAT(1H1,//////////,40X,*Y O U   G O O F E D *///1X,*YOU COMMITT
     1ED A NO-NO AND SET ISWEEPX AND ISWEEPY EQUAL TO ZERO*///,1X,*THE C
     2ALCULATION CANT PROCEED*)
  778 FORMAT(1H1,////////////1X,*CALCULATION SWEEPS WILL BE PERFORMED IN
     1BOTH X AND Y DIRECTIONS*)
  779 FORMAT(1H1,////////////1X,*CALCULATION SWEEPS WILL BE PERFORMED IN
     1THE X-DIRECTION ONLY - IMPLICIT IN Y*)
  780 FORMAT(1H1,////////////1X,*CALCULATION SWEEPS WILL BE PERFORMED IN
     1THE Y-DIRECTION ONLY - IMPLICIT IN X*)
  781 FORMAT(1H1,10(/),9X,*A R P A   I C W A K E   P R O G R A M - J W B
     1 L   C O D E*///9X,*W. J. G R A B O W S K I*,10X,A10,6X,A10,6(/)
     2,9X,*NUMERICAL SOLUTION OF INCOMPRESSIBLE,AXISYMMETRIC NAVIER-STOK
     3ES EQUATIONS*//9X,*FOR SWIRLING FLOWS WITH LARGE AXIAL GRADIENTS*,
     64(/),9X
     2,*CENTERED-UPWIND DIFFERENCING USING VARIABLE ARTIFICAL VISCOSITY*
     3,//,9X,*DIRECT SOLUTION FOR PRESSURE*//9X,*PARABOLIC OUTFLOW BOUND
     4ARY CONDITION*)
783   FORMAT(1H0,*SEQUENCE NUMBER =*,I5)
  785 FORMAT(1H0,//,* COORDINATE TRANSFORMATION PARAMETERS*,/1X,*NC =*,
     1I5,5X,*MC =*,I5,   5X,*ZCC =*,E11.4,5X,*RCC =*,E11.4,5X,*ZINITL =*
     2,E11.4/1X,*AX =*,E11.4,5X,*BX =*,E11.4,5X,*AY =*,E11.4,5X,*BY =*,
     3E11.4,5X,*EPS =*,E11.4)
786   FORMAT(1H0,*ZMAX =*,E11.4,5X,*RMAX =*,E11.4)
787   FORMAT(1H0,*XMAX =*,E13.6,10X,*YMAX =*,E13.6)
791   FORMAT(1H0,*MAX NUMBER OF ITERATIONS IN THIS RUN, NTMX =*,I5,5X,*
     1PRINT INTERVAL, NPRNT =*,I6)
 792   FORMAT(1H0,*RADIAL PRINT PARAMETER*,5X,*N1 =*,I4,5X,*N2 =*,I4,5X,
     1*N3 =*,I4)
```

ICWAKE

```
 793 FORMAT(1H0,*GRID POINTS IN X, N =*,I5,5X,*GRID POINTS IN Y, M =*,
    1 I5)
795   FORMAT(1H0,*GRID SIZE IN X, H =*,E13.6,5X,*GRID SIZE IN Y, K =*,E1
    13.6)
 799 FORMAT(1H1,/,* Z TO X TRANSFORMATION*,/5X,*I*,11X,*X*,17X,*Z*,17X,
    1*XH*,16X,*ZH*)
 800 FORMAT(1H1,*R TO Y TRANSFORMATUON*,/5X,*J*,11X,*Y*,17X,*R*,17X,*RH
    1*)
863   FORMAT(1X,I3,9F14.4)
866   FORMAT(1H1,*J =*,I4,5X,*R =*,F11.6//1X,*I,Z,U,V,W,PSI,DXP,DYP,ZH,P
    1*//1X,I3,9F14.4)
7001 FORMAT(/1X,*MASS FLUX CALCULATION*,/,1X,*WEST FLUX =*,E14.6,5X,
    1*EAST FLUX =*,E14.6,5X,*NORTH FLUX =*,E14.6,/5X,*NET OUT-FLUX =*,
    2E14.6)
 782 FORMAT(1H1,9(/),10X,*THE FLOW REYNOLDS NUMBER IS BASED ON A CHARAC
    1TERISTIC RADIUS, E.G. BODY OR NOZZLE RADIUS*/10X,*AND A CHARACTERI
    2STIC AXIAL MEAN VELOCITY, E.G. THE FREE-STREAM VELOCITY*//10X,*IN
    3THIS CALCULATION RE =*,E11.4)
 794 FORMAT(1H0,3(/),10X,*THE EQUATION SYSTEM WILL BE MARCHED WITH TIME
    1 STEP TA =*,E11.4)
7844 FORMAT(1H0,6(/),10X,*R AND Z ARE RADIAL AND AXIAL COORDINATES*/
    210X,*NON-DIMENSIONALIZED BY THE CHARACTERISTIC LENGTH*)
7845 FORMAT( / ,    10X,*U, V, AND W ARE VELOCITY IN THE RADIAL, CIRCUMF
    1ERENTIAL AND AXIAL DIRECTIONS*/10X,*NON-DIMENSIONALIZED BY THE CHA
    2RACTERISTIC VELOCITY*)
7846 FORMAT(      /,10X,*P IS PRESSURE NORMALIZED BY ITS VALUE AT POINT
    1 N,M*)
7847 FORMAT(      /,10X,*DIV IS THE DIVERGENCE OF THE VELOCITY FIELD*)
7848 FORMAT(1H0,//,10X,*IN CALCULATIONS WITH TURBULENCE, T-TERMS SUCH A
    1S TRR,TTT,TZZ REPRESENT*/10X,*TURBULENT CORRELATIONS I.E. NEGATIVE
    2 REYNOLDS STRESSES*)
8787 FORMAT(1H1)
8786 FORMAT(1H1,20(/),10X,*CONVERGENCE AFTER*,I5,3X,*ITERATIONS*/)
      STOP
      END
```

BLAYER

```
      SUBROUTINE BLAYER
      COMMON /2/MPOINT,RT(32),UL(32),VL(32),WL(32),UU(32),VV(32),WW(32),
     X          WU(32),WV(32),UV(32),EPSIL(32),ISCHETZ,IDEP,IBLPRP
      COMMON /BLP/RE,BRD,BLN,BU,BPROP,ALAMB,NB,NBEL,BELR(20),BELC(20),
     X            BELANG(20),BELTH(20),BELDEL(20),DW(32)
      DIMENSION YSDEL(12),F1(12),F3(12),F13IR(2)
      DATA YSDEL/0.,.1,.2,.3,.4,.5,.6,.7,.8,.9,1.,1.1/,
     X     F1/1.83,1.82,1.78,1.71,1.62,1.48,1.31,1.12,.88,.60,.31,0./,
     X     F3/8.65,8.65,8.65,8.65,8.46,8.17,7.50,6.25,4.23,2.12,.58,0.,0./,
     X     UL,VL,WL,UU,VV,WW,WU,WV,UV,EPSIL
     X     /32*0.,32*0.,32*1.,32*0.,32*0.,32*0.,32*0.,32*0.,32*0.,32*0./
      DATA PI/3.1415926535898/
C
C  CALCULATE VARIOUS PARAMETERS
      YSDMX=YSDEL(12)
      AR=.5*BLN/BRD
      REL=2.*RE*AR
      ALGREL=ALOG10(REL)
      CBIGF=.455/ALGREL**2.58
      CLCF=(2.*ALGREL-.65)**(-2.3)
      UTAU=SQRT(.5*CLCF)
      ARG=SQRT(AR*AR-1.)/AR
      SA=2.*PI*(1.+(AR/ARG)*ASIN(ARG))
      XN=5.
      IF(RE .GT. 1.E+6) XN=7.
      XNR=(XN+1.)/XN
      BLDEL=SQRT(1.27*SA*XNR*CBIGF)
C
C  CALCULATE AXIAL VELOCITY AND TURBULENCE
C
      IPT=-1
      DO 100 IR=1,MPOINT
      RSBRD=RT(IR)
      RSDEL=RSBRD/BLDEL
      WLIR=RSDEL**(2./XN)
      IF(WLIR .GT. 1.) WLIR=1.
      WL(IR)=WLIR
      IF(RSDEL .GT. YSDMX) GO TO 100
      CALL IUNI(12,12,YSDEL,2,F1,1,RSDEL,F13IR,IPT,IERR)
      F1IR=F13IR(1)
      F3IR=F13IR(2)
      SQK=(F1IR*UTAU)**2
      RZZ=1.134*SQK
      WW(IR)=RZZ
      UU(IR)=.328*RZZ
      VV(IR)=.436*RZZ
      RRZ=.147*SQK*(1.023-RSDEL)
      IF(RRZ .LT. 0.) RRZ=0.
      WU(IR)=-RRZ
      EPSIL(IR)=F3IR*UTAU**3/BLDEL
  100 CONTINUE
C
      PRINT 101, (RT(I),WL(I),WW(I),UU(I),VV(I),WU(I),
     X          EPSIL(I), I=1,MPOINT)
  101 FORMAT(1H1,*CALCULATION OF AXIAL VELOCITY AND TURBULENCE AT PROPEL
     XLER ENTRANCE*//////)
```

```
MLAYER
              X        1H ,8X,*RT*,13X,*WL*,13X,*WW*,13X,*UU*,13X,*VV*,13X,*WU*,
              X        11X,*EPSIL*,//(1H ,7E15.5))
          C
                RETURN
                END
```

PROPWV

```
      SUBROUTINE PROPWV
      COMMON /2/MPOINT,RT(32),UL(32),VL(32),WL(32),UU(32),VV(32),WW(32),
     X          WU(32),WV(32),UV(32),EPSIL(32),ISCHETZ,IDEP,IALPRP
      COMMON /BLP/RE,BRD,BLN,BU,RPROP,ALAMB,NB,NBEL,BELR(20),BELC(20),
     X             BELANG(20),BELTH(20),BELDEL(20),DW(32)
      DIMENSION DWB(22),VLB(22),WLB(21),DWVLI(2)
      DATA PI/3.14159265358979/
C
      PRINT 1
    1 FORMAT(1H1,*CALCULATION OF AXIAL AND SWIRL VELOCITIES AT PROPELLER
     X EXIT*)
C
C  INTERPOLATE WL FROM RT TO BELR
      NBP1=NBEL+1
      BELR(NBP1)=RPROP
      IPT=-1
      DO 10 IB=1,NBP1
      BELRAD=BELR(IB)/BRD
      CALL IUNI(32,MPOINT,RT,1,WL,1,BELRAD,WLI,IPT,IERR)
      WLB(IB)=WLI
   10 CONTINUE
C
C  CALCULATE DWB,VLB,WLB AT PROPELLER EXIT
C
      VINF=BU
      VT=WLB(NBP1)*VINF
      EFLAMT=ALAMB*VT/VINF
      DO 100 IB=1,NBEL
      R=BELR(IB)
      C=BELC(IB)
      THETA=BELANG(IB)
      THICK=BELTH(IB)
      DELTA=BELDEL(IB)
      X=R/RPROP
      V=VINF*WLB(IB)
      EFLAM=ALAMB*V/VINF
      VBE=V*SQRT(1.+(X/EFLAM)**2)
C  KINEMATIC VISCOSITY = .01 CM*CM/SEC
      REC=VBE*C/.01
C
C  THICKNESS EFFECT ON BLADE ELEMENT DRAG COEFFICIENT
      ALR=ALOG10(REC)
      IF(ALR .LT. 5.652) GO TO 30
      ALCD0=-2.4
      GO TO 31
   30 ALCD0=-1.574-.5*(ALR-4.)
   31 IF(ALR .GT. 5.) GO TO 35
      ALCD25=-1.-.2*(ALR-4.)
      GO TO 40
   35 IF(ALR .GE. 6.) GO TO 37
      ALCD25=-1.2-.93*(ALR-5.)
      GO TO 40
   37 ALCD25=-2.13
   40 CONTINUE
      TC=THICK/C
      CDBL=10.**(ALCD0*(1.-4.*TC)+4.*TC*ALCD25)
```

PROPWV

```
      RNB=FLOAT(NB)
      SIG=RNB*C/(2.*PI*R)
C
C  AZERO IS LIFT CURVE SLOPE
      AZERO=2.*PI/(.924+2.22*PI*PI*CDRL)
C
C  CORRECTION DUE TO BLADE THICKNESS
      DENOM=EFLAM*EFLAM+X*X
      EFTHET=PI*THETA/180.-4./15.*TC*EFLAM*X*SIG/DENOM
C
C  ZERO LIFT ANGLE DUE TO GEOMETRIC CAMBER
      THETEF=ATAN(2.*DELTA/C)+EFTHET
      PHIG=ATAN(EFLAM/X)
C
C  PRANDTL F-FACTOR FOR TIP CORRECTION
      FSMALL=(RNB/2.)*(1.-X)*SQRT(1.+1./(EFLAMT*EFLAMT))
      FPRAND=(2./PI)*ACOS(EXP(-FSMALL))
C
C  INDUCED ANGLE OF ATTACK
C  (ITERATIONS TO INCLUDE EFFECTIVE BLADE CURVATURE)
      K=0
   50 CONTINUE
      T1=4.*SIN(PHIG)*FPRAND/(SIG*AZERO)+1.
      T2=16.*FPRAND*COS(PHIG)*(THETEF-PHIG)/(SIG*AZERO)
      ALFIND=2.*(THETEF-PHIG)/(T1+SQRT(T1*T1+T2))
      ALFA=THETEF-PHIG-ALFIND
      CL=AZERO*ALFA
C
C  ACTUAL DRAG COEFF BASED ON DATA BY HOERNER
      CDDL=CDBL*(1.+2.*CL*CL)
      PHI=PHIG+ALFIND
      CX=CL*SIN(PHI)+CDDL*COS(PHI)
      CY=CL*COS(PHI)-CDDL*SIN(PHI)
      AK=SIG*CX/(2.*SIN(2.*PHI))
      APRIME=AK/(1.+AK)
      AKBAR=SIG*CY/(4.*SIN(PHI)**2)
      AA=AKBAR/(1.-AKBAR)
      ARG=EFLAM*(1.+AA)/X
      DI=ABS(THETEF)
      THETEF=EFTHET+.25*(ATAN(ARG/(1.-2.*APRIME))-ATAN(ARG))
      DII=ABS(THETEF)
      DIFF=ABS((DII-DI)/DI)
      K=K+1
C
C  CONVERGENCE TEST
      IF(DIFF .LE. .001) GO TO 70
      IF(K.LT.50) GO TO 50
      PRINT 61, IB
   61 FORMAT(/////1H ,*NO CONVERGENCE IN 50 ITERATIONS FOR IB = *,I2/
     X        1H ,*CHECK GEOMETRIC CAMBER INPUT*)
C
C  OBTAIN VS,DV
   70 CONTINUE
      DV=AA*V/VINF
      VS=2.*X*APRIME/ALAMB
      VS=VS*V/VINF
```

PROPWV

```
      DWB(IB)=DV
      VLR(IB)=VS
  100 CONTINUE
C
C
C  MAKE ROOM FOR AXIS
      DO 110 JB=1,NBEL
      IB=NBEL-JB+1
      IBP1=IB+1
      BELR(IBP1)=BELR(IB)
      DWB(IBP1)=CWR(IB)
      VLB(IBP1)=VLR(IB)
  110 CONTINUE
C  ASSIGN DWB,VLB = 0. AT AXIS AND TIP
      BELR(1)=0.
      DWB(1)=0.
      VLR(1)=0.
      NBEL=NBEL+2
      BELR(NBEL)=RPROP
      DWB(NBEL)=0.
      VLB(NBEL)=0.
C
C  INTERPOLATE DWB,VLB FROM BELR TO RT, ADJUST WL
      IPT=-1
      DO 120 IR=1,MPOINT
      DW(IR)=0.
      VL(IR)=0.
      RTI=RT(IR)*BRD
      IF(RTI.GT.RPROP) GO TO 120
      CALL IUNI(22,NBEL,BELR,2,DWR,1,RTI,DWVLI,IPT,IERR)
      DW(IR)=DWVLI(1)
      VL(IR)=DWVLI(2)
      WL(IR)=WL(IR)+DW(IR)
  120 CONTINUE
C
      PRINT 121, (RT(I),DW(I),VL(I),WL(I), I=1,MPOINT)
  121 FORMAT(/////1H ,14X,*RT*,18X,*DW*,18X,*VL*,18X,*WL*//(1H ,4E20.5))
C
      RETURN
      END
```

PROPU

```
      SUBROUTINE PROPU
      COMMON /2/MPOINT,RT(32),UL(32),VL(32),WL(32),UU(32),VV(32),WW(32),
     X           WU(32),WV(32),UV(32),EPSIL(32),ISCHETZ,IDEP,IDLPRP
      COMMON /BLP/RE,BRD,BLN,BU,RPROP,ALAMB,NR,NBEL,BELR(20),BELC(20),
     X           BELANG(20),BELTH(20),BELDEL(20),DW(32)
      DIMENSION QGRND1(401),QGRND2(401),RHOM(401),RHOP(401)
      EQUIVALENCE (RHOM,RHOP)
      DATA PI/3.1415926535898/
C
   21 FORMAT(1H1,*CALCULATION OF RADIAL VELOCITY AT PROPELLER EXIT*/
     X       /////1H ,*MLIM,RLIM = *,I10,E20.5)
   41 FORMAT(/////1H ,*IR,R,RM,RP = *,I10,3E20.5)
  101 FORMAT(1H0,*Q1M1,Q1M2,Q2M = *,3E20.5)
  151 FORMAT(1H0,*Q1P1,Q1P2,Q2P = *,3E20.5)
  161 FORMAT(1H0,*UL(IR) = *,E20.5)
  171 FORMAT(/////1H ,*UL(MLIM) = *,E20.5)
C
C  FIND POINT WHERE AXIAL VELOCITY EQUALS FREESTREAM VALUE
      DO 10 I=2,MPOINT
      IF(ABS(DW(I)) .GT. 1.E-10) GO TO 10
      MLIM=I
      GO TO 20
   10 CONTINUE
C
   20 CONTINUE
      RLIM=RT(MLIM)
      PRINT 21, MLIM,RLIM
      IF(MLIM.GE.MPOINT .OR. RLIM.GE. 2.) STOP
      UL(1)=0.
      GFAC=200.
      DSING=.01/GFAC
C
C  COMPUTE NON-ZERO VALUES OF RADIAL VELOCITY
C
      DO 200 IR=2,MPOINT
      IF(IR.EQ.MLIM) GO TO 200
      R=RT(IR)
      RM=AMIN1(R-DSING,RLIM)
      RP=R+DSING
      PRINT 41, IR,R,RM,RP
C
C  COMPUTE MINUS PART OF SINGULAR INTEGRALS
      NRHOM=IFIX(GFAC*RM+.1) + 1
      DRHOM=RM/FLOAT(NRHOM-1)
      DO 90 I=1,NRHOM
   90 RHOM(I)=DRHOM*FLOAT(I-1)
C
      IPT=-1
      DO 100 IRHC=1,NRHOM
      RHO=RHOM(IRHO)
      CALL IUNI(32,MPOINT,RT,1,DW,1,RHC,DWI,IPT,IERR)
      XM=SQRT(4.*R*RHO/(R+RHO)**2)
      VALK=ELINT1(XM)
      VALE=ELINT2(XM)
      QGRND1(IRHC)=DWI*VALK*RHO/(R+RHO)
      QGRND2(IRHC)=DWI*VALE*RHO/(R-RHO)
```

PROPU

```
  100 CONTINUE
      CALL QSF(DRHOM,QGRND2,QGRND2,NRHCM)
      Q2M=QGRND2(NRHOM)
C
      Q1M2=0.
      IF(IR.LT.MLIM) GC TO 105
      CALL QSF(DRHOM,QGRND1,QGRND1,NRHCM)
      Q1M1=QGRND1(NRHOM)
      GO TO 115
  105 NRMM1=NRHOM-1
      CALL QSF(DRHOM,QGRND1,QGRND1,NRMM1)
      Q1M1=QGRND1(NRMM1)
C
      RMM1=RHOM(NRMM1)
      NRHOM=21
      DRHOM=(RM-RMM1)/FLOAT(NRHOM-1)
      IPT=-1
      DO 110 IRHO=1,NRHOM
      RHO=RMM1+DRHOM*FLOAT(IRHO-1)
      CALL IUNI(32,MPOINT,RT,1,DW,1,RHC,DWI,IPT,IERR)
      XM=SQRT(4.*R*RHO/(R+RHO)**2)
      VALK=ELINT1(XM)
      QGRND1(IRHO)=DWI*VALK*RHO/(R+RHO)
  110 CONTINUE
      CALL QSF(DRHOM,QGRND1,QGRND1,NRHCM)
      Q1M2=QGRND1(NRHOM)
C
  115 PRINT 101, Q1M1,Q1M2,Q2M
C
C  COMPUTE PLUS PART OF SINGULAR INTEGRALS
      Q1P1=0.
      Q1P2=0.
      Q2P=0.
      RLMMRP=RLIM-RP
      IF(IR.GT.MLIM) GC TO 165
      NRHOP=IFIX(GFAC*RLMMRP+.1) + 1
      DRHOP=RLMMRP/FLOAT(NRHOP-1)
      DO 140 I=1,NRHOP
  140 RHOP(I)=RP + DRHCP*FLOAT(I-1)
C
      IPT=-1
      DO 150 IRHO=1,NRHOP
      RHO=RHOP(IRHO)
      CALL IUNI(32,MPOINT,RT,1,DW,1,RHC,DWI,IPT,IERR)
      XM=SQRT(4.*R*RHO/(R+RHO)**2)
      VALK=ELINT1(XM)
      VALE=ELINT2(XM)
      QGRND1(IRHC)=DWI*VALK*RHO/(R+RHO)
      QGRND2(IRHC)=DWI*VALE*RHO/(R-RHO)
  150 CONTINUE
      CALL QSF(DRHOP,QGRND2,QGRND2,NRHOP)
      Q2P=QGRND2(NRHOP)
C
      NRPM1=NRHOP-1
      CALL QSF(DRHOP,QGRND1(2),QGRND1(2),NRPM1)
      Q1P1=QGRND1(NRHOP)
```

```
C
      RLM2=RHOP(2)
      NRHOP=21
      DRHOP=(RLM2-RP)/FLOAT(NRHOP-1)
      IPT=-1
      DO 160 IRHO=1,NRHOP
      RHO=RP+DRHOP*FLOAT(IRHO-1)
      CALL IUNI(32,MPOINT,RT,1,DW,1,RHO,DWI,IPT,IERR)
      XM=SQRT(4.*R*RHO/(R+RHO)**2)
      VALK=ELINT1(XM)
      QGRND1(IRHO)=DWI*VALK*RHO/(R+RHO)
  160 CONTINUE
      CALL QSF(DRHOP,QGRND1,QGRND1,NRHOP)
      Q1P2=QGRND1(NRHOP)
C
  165 PRINT 151, Q1P1,Q1P2,Q2P
C
C  CALCULATE RADIAL VELOCITY
      UL(IR)=-(Q1M1+Q1M2+Q1P1+Q1P2+Q2M+Q2P)/(PI*G)
      PRINT 161, UL(IR)
  200 CONTINUE
      RRLP1=(RLIM-RT(MLIM-1))/(RT(MLIM+1)-RT(MLIM-1))
      RRLM1=1.-RRLP1
      UL(MLIM)=RRLP1*UL(MLIM+1) + RRLM1*UL(MLIM-1)
      PRINT 171, UL(MLIM)
C
      RETURN
      END
```

PTURB

```
      SUBROUTINE PTURB
      COMMON /2/MPOINT,RT(32),UL(32),VL(32),WL(32),UU(32),VV(32),WW(32),
     X           WU(32),WV(32),UV(32),EPSIL(32),ISCHETZ,IDEP,IBLPRP
C
C  APPLY PROPELLER JUMP CONDITIONS TO TURBULENCE
C
      DO 10 IR=2,MPOINT
      VSW=VL(IR)/WL(IR)
      VSWSQ=VSW*VSW
      SQK=.5*(WW(IR)+VV(IR)+UU(IR))
      FOFR=-(.327*WW(IR)-.0182*VV(IR)+.0909*UU(IR))
      VWF=VSWSQ*FOFR
      WW(IR)=WW(IR)-.364*VWF
      VV(IR)=VV(IR)-1.06*VWF
      UU(IR)=UU(IR)-.582*VWF
      WU(IR)=WU(IR)*(1.-.0258*VSWSQ)
      WV(IR)=VSW*FOFR
      UV(IR)=-.236*VSW*WU(IR)
      IF(SQK.LT.1.E-10)GO TO 10
      EPSIL(IR)=EPSIL(IR)*(1.-1.44*VWF/SQK)
   10 CONTINUE
C
C  CORRECT FOR DISCONTINUITY IN SHEAR STRESS BOUNDARY CONDITION
      WU(1)=0.
      WU(2)=.5*WU(2)
C
      PRINT 11, (RT(I),WW(I),VV(I),UU(I),WU(I),WV(I),UV(I),
     X            EPSIL(I), I=1,MPOINT)
   11 FORMAT(1H1,*CALCULATION OF TURBULENCE AT PROPELLER EXIT*//////
     X        1H ,8X,*RT*,13X,*WW*,13X,*VV*,13X,*LU*,
     X        13X,*WU*,13X,*WV*,13X,*UV*,11X,*EPSIL*//(1H ,8E15.5))
      PRINT 13
   13 FORMAT(1H1)
C
      RETURN
      END
```

```
ELINT1
      FUNCTION ELINT1(XK)
C
C        THIS ROUTINE SOLVES COMPLETE ELLIPTIC INTEGRALS OF THE FIRST
C        KIND BY USING CHEBYSHEV APPROXIMATIONS. THE MAXIMAL ERROR
C        IS 1.99E-13.
C
      DIMENSION A(8), B(8)
      DATA( A(I), I=1,8 ) / 1.38629436111989E+00, 9.65736020516771E-02,
     13.08909633861795E-02, 1.52618320622534E-02, 1.25565693543211E-02,
     21.68695685967517E-02, 1.09423810688623E-02, 1.40704915496101E-03 /
      DATA( B(I), I=1,8 ) / .5, 1.24999998585309E-01,
     17.03114105853296E-02, 4.87379510945218E-02, 3.57218443007327E-02,
     22.09857677336790E-02, 5.81807961871996E-03, 3.42805719229748E-04 /
      X=1.-XK*XK
      SUM1= A(8)
      SUM2= B(8)
      DO1 I=1,7
      J = 8-I
      SUM1= X*SUM1+ A(J)
      SUM2= X*SUM2+ B(J)
    1 CONTINUE
      ELINT1=SUM1-ALOG(X)*SUM2
      RETURN
      END
```

ELINT2

```
      FUNCTION ELINT2(XK)
C
C        THIS ROUTINE SOLVES COMPLETE ELLIPTIC INTEGRALS OF THE SECOND
C        KIND BY USING CHEBYSHEV APPROXIMATIONS. THE MAXIMAL ERROR
C        IS 2.18E-13.
C
      DIMENSION A(8), B(8)
      DATA( A(I), I=1,8 ) / 1.0, 4.43147193467733E-01,
     15.68115681053803E-02, 2.21862206993846E-02, 1.56847700239786E-02,
     21.92204389022977E-02, 1.21819481486695E-02, 1.55618744745296E-03 /
      DATA( B(I), I=1,8 ) / 0.0, 2.49999998448655E-01,
     19.37488062098189E-02, 5.84950297066166E-02, 4.09074821593164E-02,
     22.35091602564984E-02, 6.45682247315060E-03, 3.78886487349367E-04 /
      X=1.-XK*XK
      SUM1= A(8)
      SUM2= B(8)
      DO1 I=1,7
      J = 8-I
      SUM1= X*SUM1+ A(J)
      SUM2= X*SUM2+ B(J)
    1 CONTINUE
      ELINT2=SUM1-ALOG(X)*SUM2
      RETURN
      END
```

```
QSF
      SUBROUTINE QSF(H,Y,Z,NDIM)
C
C     H      -THE INCREMENT OF ARGUMENT VALUES.
C     Y      -THE INPUT VECTOR OF FUNCTION VALUES.
C     Z      -THE RESULTING VECTOR OF INTEGRAL VALUES. Z MAY BE
C             IDENTICAL WITH Y.
C     NDIM   -THE DIMENSION OF VECTORS Y AND Z.
C
      DIMENSION Y(1),Z(1)
C
      HT = .3333333*H
      IF(NDIM-5)7,8,1
C
C     NDIM IS GREATER THAT 5. PREPARATIONS OF INTEGRATION LOOP
    1 SUM1=Y(2)+Y(2)
      SUM1=SUM1+SUM1
      SUM1=HT*(Y(1)+SUM1+Y(3))
      AUX1=Y(4)+Y(4)
      AUX1=AUX1+AUX1
      AUX1=SUM1+HT*(Y(3)+AUX1+Y(5))
      AUX2=HT*(Y(1)+3.875*(Y(2)+Y(5))+2.625*(Y(3)+Y(4))+Y(6))
      SUM2=Y(5)+Y(5)
      SUM2=SUM2+SUM2
      SUM2=AUX2-HT*(Y(4)+SUM2+Y(6))
      Z(1) = 0.
      AUX=Y(3)+Y(3)
      AUX=AUX+AUX
      Z(2)=SUM2-HT*(Y(2)+AUX+Y(4))
      Z(3)=SUM1
      Z(4)=SUM2
      IF(NDIM-6)5,5,2
C
C     INTEGRATION LOOP
    2 DO 4 I=7,NDIM,2
      SUM1=AUX1
      SUM2=AUX2
      AUX1=Y(I-1)+Y(I-1)
      AUX1=AUX1+AUX1
      AUX1=SUM1+HT*(Y(I-2)+AUX1+Y(I))
      Z(I-2)=SUM1
      IF(I-NDIM)3,6,6
    3 AUX2=Y(I)+Y(I)
      AUX2=AUX2+AUX2
      AUX2=SUM2+HT*(Y(I-1)+AUX2+Y(I+1))
    4 Z(I-1)=SUM2
    5 Z(NDIM-1)=AUX1
      Z(NDIM)=AUX2
      RETURN
    6 Z(NDIM-1)=SUM2
      Z(NDIM)=AUX1
      RETURN
C     END OF INTEGRATION LOOP
C
    7 IF(NDIM-3)12,11,8
C
C     NDIM IS EQUAL TO 4 OR 5
```

OSF

```
    8 SUM2=1.125*HT*(Y(1)+Y(2)+Y(2)+Y(2)+Y(3)+Y(3)+Y(3)+Y(4))
      SUM1=Y(2)+Y(2)
      SUM1=SUM1+SUM1
      SUM1=HT*(Y(1)+SUM1+Y(3))
      Z(1)=0.
      AUX1=Y(3)+Y(3)
      AUX1=AUX1+AUX1
      Z(2)=SUM2-HT*(Y(2)+AUX1+Y(4))
      IF(NDIM-5)10,9,9
    9 AUX1=Y(4)+Y(4)
      AUX1=AUX1+AUX1
      Z(5)=SUM1+HT*(Y(3)+AUX1+Y(5))
   10 Z(3)=SUM1
      Z(4)=SUM2
      RETURN
C
C     NDIM IS EQUAL TO 3
   11 SUM1=HT*(1.25*Y(1)+Y(2)+Y(2)-.25*Y(3))
      SUM2=Y(2)+Y(2)
      SUM2=SUM2+SUM2
      Z(3)=HT*(Y(1)+SUM2+Y(3))
      Z(1)=0.
      Z(2)=SUM1
   12 RETURN
      END
```

```
COEFF
      SUBROUTINE COEFF(ZINITL,RCC,ZCC)
C
C     .....SUBROUTINE CCEFF COMPUTES ANC STORES COEFFICIENTS WHICH ARE
C     .....USED REPEATECLY IN CALCULATIONS.
C
      REAL K,KS,KSI,KI,KH,K2I
      COMMON/VLIF/RP(41),JND(41)
      COMMON/DAT/X(61),Y(61),R(61),Z(61),F(61),S(61),S2(61),C1(61)
     1,C2(61),C3(61),C4(61),C5(61),C6(61),C7(61),C10(61),C11(61),C12(61)
     2, TA,RE,H,K,N,M,AM,MM,MI2,HSI,KSI,HH,KH,E(61),F2(61),ZM(61),XM(61)
     3, AX,BX,AY,BY,NUMBER,NC,MC,EPS,NSTRT,GEF(61),TH(61),ITURB,NTURB
      COMMON/PRESCO/ AM(61),AN(61),BM(61),BN(61),CM(61),CN(61),EM(61),
     1  EMS(61),FM(61),FMS(61),RM(61),SM(61),SMS(61),THNM,WORK(600),
     2  BCO1,BCO2,BCO3,SI,TI,SIS,SIREI,HH2,IFLG,H2I,K2I
      COMMON/VEL/U(61,41),V(61,41),W(61,41),P(61,41),DIV(61,41)
      COMMON/UPSTRM/WH(61),WHRR(61),WI,VI,ALPH,RMAX,ZMAX,REI,THETA1,RR

      IFLG = 0
      HH = 0.5*H
      KH = 0.5*K
      KS = K*K
      HS = H*H
      KSI = 1./KS
      HSI = 1./HS
      KI = 1./K
      HI = 1./H
      TDK = TA*KI
      TDH = TA*HI
      H2I = 0.5*HI
      K2I = 0.5*KI
      REI = 1./RE
      TAI=1./TA
C     .....LOGARITHMIC TRANSFORMATIONS
      ZMZ=ZMAX-ZINITL
      RAT=ZMZ/(ZCC-ZINITL)
      CALL NEWT(AX,BX,RAT,ZMZ,NC,N,EPS,X(N))
      RAT=RMAX/RCC
      CALL NEWT(AY,BY,RAT,RMAX,MC,M,EPS,Y(M))
      DO 52J = 1,M
      YJ = (J-1)*K
      Y(J) = YJ
      RJ = BY*(EXP(AY*YJ)-1.)
      R(J) = RJ
      FJ = (EXP(-AY*YJ))/(AY*BY)
      GJ = -AY*FJ*FJ
      IF(J.NE.1) GO TO 40
      EJ = 0.
      E(J) = 0.
      GO TO 41
   40 CONTINUE
      EJ = 1./RJ
      E(J)=EJ
   41 CONTINUE
      F(J)=.5*FJ*KI
      F2(J) = F(J)*2.
      A1=FJ*FJ*KSI*REI
```

```
COEFF
      A2=(GJ+EJ*FJ)*.5*KI*REI
      C4(J) = A1-A2
      C6(J) = A1+A2
      C5(J)=2.*TA1-2.*A1
      C7(J)=C5(J)-EJ*EJ*REI
      C11(J)=4.*TA1-C5(J)
   52 C12(J)=C11(J)+EJ*EJ*REI

      DO 53I = 1,N
      XI = (I-1)*H
      X(I) = XI
      Z(I) = BX*(EXP(AX*XI)-1.)+ZINITL
      SI = (EXP(-AX*XI))/(AX*BX)
      TI = -AX*SI*SI
      S(I)=0.5*SI*HI
      S2(I) = S(I)*2.
      A1=SI*SI*HSI*REI
      A2=TI*.5*HI*REI
      C1(I) = A2-A1
      C2(I)=2.*TA1+2.*A1
      C3(I) = -A2-A1
   53 C10(I)=4.*TA1-C2(I)
C       PRESSURE COEFFICIENTS
      MM2 = M-2
      DO 61J = 1,MM
      JP = J+1
      YH = (FLOAT(J)-0.5)*K
      RH(J) = BY*(EXP(AY*YH)-1.)
      FHJ = EXP(-AY*YH)/(AY*BY)
      GHJ = -AY*FHJ*FHJ
      FH(J) = FHJ*KI*0.5
      FHS(J) = FH(J)*FH(J)
      EHJ = 1./RH(J)
      EH(J) = 0.5*EHJ
      EHS(J) = 0.25*EH(J)*EH(J)
      HHJ = (FHJ*EHJ+GHJ)*0.5*KI
      GEF(J) = 2.*HHJ
      TEM1 = FHJ*FHJ*KSI
      IF(J.GT.1)GO TO 64
      AM(1) = 0.
      BM(1) = -TEM1-HHJ
      CM(1) = TEM1+HHJ
      GO TO 61
   64 IF(J.LT.MM) GO TO 65
      AM(MM) = TEM1-HHJ
      BM(MM) = -TEM1+HHJ
      CM(MM) = 0.
      GO TO 61
   65 AM(J) = TEM1-HHJ
      BM(J) = -2.*TEM1
      CM(J) = TEM1+HHJ
   61 CONTINUE
C
      DO 71I = 1,NM
      XM(I) = (FLOAT(I)-0.5)*H
```

COEFF

```
      ZH(I) = BX*(EXP(AX*XH(I))-1.)+ZINITL
      SHI = EXP(-AX*XH(I))/(AX*BX)
      SH(I) = 0.5*HI*SHI
      THI = -AX*SHI*SHI
      SHS(I) = SH(I)*SH(I)
      TEM2 = 0.5*THI*HI
      TH(I) = 2.*TEM2
      IF(I.EQ.NN)THNN=TEM2
      TEM1 = SHI*SHI*HSI
      IF(I.GT.1) GO TO 74
      AN(I) = 0.
      BN(I) = -TEM1-TEM2
      CN(I) = TEM1+TEM2
      GO TO 71
   74 IF(I.LT.NN) GO TO 75
      AN(NN) = 2.*TEM1-TEM2
      BN(NN) = -6.*TEM1-TEM2
      CN(NN) = 0.
      GO TO 71
   75 AN(I) = TEM1-TEM2
      BN(I) = -2.*TEM1
      CN(I) = TEM1+TEM2
   71 CONTINUE

      S1=2.*S(1)*H
      T1 = -AX*S1*S1
      SH1 = 2.*H*SH(1)
      TH1 = -AX*SH1*SH1
      BC01 = SH1*SH1*HI-0.5*TH1
      BC02 = -SH1*SH1*HI-0.5*TH1
      S1S = S1*S1
      S1REI = 1./(S1*RE)
      FH=2.*F(H)*K
      FHH = 2.*K*FH(NN)
      GHH = -AY*FHH*FHH
      EHH = 1./RH(NN)
      BC03 = -FHH*FHH*KI-0.5*(GHH+EHH*FHH)
      RETURN
      END
```

```
NEWT
          SUBROUTINE NEWT(X,Y,R,RR,NC,N,EPS,A)
          H = A/(N-1)
          P = (NC-1)*H
    10    P1 = EXP(A*X)-1.
          P2 = EXP(P*X)-1.
          F = (P1/P2)-R
          DF = A*(P1+1.)/P2
          DF = DF-P*(P2+1.)*P1/P2**2
          XN = X-F/DF
          IF(ABS(XN-X).LT.EPS)GO TO 11
          X = XN
          GO TO 10
    11    X = XN
          Y=RR/(EXP(A*X)-1.)
          RETURN
          END
```

UPCOND

```
      SUBROUTINE UPCOND(IMAGER,IOFF)
C.......................................................................
C     .....SUBROUTINE UPCOND COMPUTES AND STORES UPSTREAM AND INITIAL
C     .....CONDITIONS
C.......................................................................
      REAL K,KS,KSI,KI,KH,K2I
      COMMON/VLIF/RP(41),JND(41)
      COMMON/VEL/U(61,41),V(61,41),W(61,41),P(61,41),DIV(61,41)
      COMMON/DAT/X(61),Y(61),R(61),Z(61),F(61),S(61),S2(61),C1(61)
     1,C2(61),C3(61),C4(61),C5(61),C6(61),C7(61),C10(61),C11(61),C12(61)
     2, TA,RE,N,K,N,M,AM,MM,MI2,MSI,KSI,HH,KH,E(61),F2(61),ZH(61),XH(61)
     3, AX,BX,AY,BY,NUMBER,NC,MC,EPS,NSTRT,GEF(61),TH(61),ITURB,NTURB
      COMMON/PRESCO/ AP(61),AN(61),BM(61),BN(61),CM(61),CN(61),EW(61),
     1  EMS(61),FM(61),FMS(61),RH(61),SM(61),SMS(61),THNM,WORK(600),
     2  BC01,BC02,BC03,S1,T1,S1S,S1REI,MM2,IFLG+21,K2I
      COMMON/UPSTRM/WH(61),WHRR(61),WI,VI,ALPH,RMAX,ZMAX,REI,THETAI,RR
      COMMON/TRICAT/DU(61),DV(61),DW(61),UT(61),VT(61),WT(61)
      COMMON/STRESS/TRR(61,32),TZZ(61,32),TTT(61,32),TRT(61,32),
     1  TRT(61,32),TZT(61,32),EPSL(61,32)
      NAMELIST/DAT9/UT,VT,WT
      PI = 3.14159265
      CON = 180./PI
      IF(IMAGER.EQ.0) GO TO 30
      ALPO = ALPH
      VO = VI
      WO = WI
      IF(NSTRT.NE.0) GO TO 47
      DO 41J = 1,MM
      ET = RH(J)
      IF(ET.GT.1.)GO TO 42
      F1 = ET*ET*(6.-8.*ET+3.*ET*ET)
      WH(J) = WO*(ALPO+(1.-ALPO)*F1)
      WHRJ = WO*(1.-ALPO)*(12.-24.*ET+12.*ET*ET)
      WHRR(J) = WO*(1.-ALPO)*(12.-48.*ET+36.*ET*ET)+WHRJ
      GO TO 41
   42 WH(J) = 1.
      WHRR(J) = 0.
   41 CONTINUE
      DO 21J = 1,M
      ET = R(J)
      IF(ET.GT.1.)GO TO 22
      F1 = ET*ET*(6.-8.*ET+3.*ET*ET)
      WT(J) = WO*(ALPO+(1.-ALPO)*F1)
      VT(J) = VO*ET*(2.-ET*ET)
      UT(J) = 0.
      GO TO 21
   22 VT(J) = VO/ET
      WT(J) = WO
      UT(J) = 0.
21    CONTINUE
   47 CONTINUE
      PRINT 93
      PRINT 90,WO,VO,ALPO
      PRINT 87
      WRITE(6,DAT9)
C     .....COMPUTE MAGER FLOW-FORCE DEFICIENCY FOR MAGER*S CUBIC-QUADRATI
```

```
UPCOND
      C     .....PROFILES
            FI = 0.047619+0.004762*ALPO-0.052381*ALPO*ALPO
            RR = VI*VI
            THETA1 = 0.25*ALOG(RR)+FI/RR
            PRINT 91,RR,THETA1
      C     .....COMPUTE SWIRL ANGLES
            FDEL = ATAN(VO/WC)*CON
            IF(NSTRT.EQ.0) GO TO 31
            GO TO 46
         30 CONTINUE
            IF(NSTRT.NE.0) GO TO 62
            CALL DATIN
         62 CONTINUE
            DO 64J = 1,M
            WT(J) = W(1,J)
            VT(J) = V(1,J)
         64 UT(J) = U(1,J)
         63 CONTINUE
            PRINT 87
            WRITE(6,DAT9)
            J=1
            DWRR = 3.*WT(1)-7.*WT(2)+5.*WT(3)-WT(4)
            WHRR(1) = 2.*FMS(1)*DWRR+GEF(1)*(WT(2)-WT(1))
            WH(1) = 0.5*(WT(2)+WT(1))
            DO 56J = 2,MM2
            WH(J) = 0.5*(WT(J+1)+WT(J))
         56 WHRR(J) = 2.*FMS(J)*(WT(J+2)-WT(J)-WT(J+1)+WT(J-1))+GEF(J)*(WT(J+1
           1)-WT(J))
            J=MM
            WH(MM) = 0.5*(WT(M)+WT(MM))
            DWRR = 3.*WT(M)-7.*WT(MM)+5.*WT(MM2)-WT(M-3)
            WHRR(MM) = 2.*FMS(MM)*DWRR+GEF(MM)*(WT(M)-WT(MM))
         31 CONTINUE
            IF(NUMBER.GT.1.AND.NSTRT.EQ.0) GO TO 45
            IF(NSTRT.GT.0) GO TO 46
            RNM1=FLOAT(N-1)
            DO 20I = 1,N
            RRNMI=FLOAT(N-I)/RNM1
            RRIM1=FLOAT(I-1)/RNM1
            DO 20J = 1,M
            U(I,J)=UT(J)*RRNMI
            V(I,J)=VT(J)*RRNMI
            W(I,J)=WT(J)*RRNMI+RRIM1
            DIV(I,J) = 0.
            IF(ITURB.EQ.0) GO TO 20
            TRR(I,J) = TRR(1,J)
            TTT(I,J) = TTT(1,J)
            TZZ(I,J) = TZZ(1,J)
            TRT(I,J) = TRT(1,J)
            TRZ(I,J) = TRZ(1,J)
            TZT(I,J) = TZT(1,J)
            EPSL(I,J) = EPSL(1,J)
         20 P(I,J) = 0.
         45 CONTINUE
      C .....  COMPUTE CIILATATION AND PRESSURE
            DO 51 I=1,NM
```

UPCOND

```
      DO 51 J=1,MM
      DWZ=SH(I)*(W(I+1,J+1)+W(I+1,J)-W(I,J+1)-W(I,J))
      TEM1 = (U(I,J+1)+U(I+1,J+1))*R(J+1)
      TEM2=(U( I,J)+U(I+1,J))*R(J)
   51 DIV(I,J) = DWZ+FM(J)*(TEM1-TEM2)/RM(J)
      P(N,MM)=0.
      DO 52J = 2,MM
      L=M-J
      TEM1 = 0.5/F(L+1)*V(N,L+1)**2/R(L+1)
      IF(ITURB.EQ.0) GO TO 52
      IF(IOFF.EQ.1) GO TO 52
      TEM1 = TEM1+0.5/F(L+1)*(TTT(N,L+1)-TRR(N,L+1))/R(L+1)
      TEM1 = TEM1-0.5*(TRR(N,L+2)-TRR(N,L))
   52 P(N,L) = P(N,L+1)-TEM1
      CALL PRESSR(IOFF)
  46  CONTINUE
      FMAX=0.
      DO 33J = 1,MM
      FF = ATAN(VT(J)/WT(J))
      IF(FF.GT.FMAX) FMAX=FF
   33 CONTINUE
      FMAX = FMAX*CON
      PRINT 92,FMAX,FDEL
   87 FORMAT(1H1)
   90 FORMAT(1H0,*FREE-STREAM AXIAL VEL, WI =*,E11.4,5X,*CORE-EDGE SWIRL
     1 VEL, VI=*,E11.4,5X,*PROFILE PARAMETER, ALPH =*,E11.4)
   91 FORMAT(1H0,*PR =*,E13.6,5X,*THETA1 =*,E13.6)
   92 FORMAT(1H0,*SWIRL ANGLES(DEGREES)*,5X,*MAXIMUM =*,E11.4,5X,*CORE E
     1DGE =*,E11.4)
   93 FORMAT(1H1,10(/),10X,*MAGER"S UPSTREAM PROFILES ARE ASSUMED IN THI
     1S LAMINAR FLOW CALCULATION*//)
      RETURN
      END
```

```
DATIN
      SUBROUTINE DATIN
C..........................................................................
C     .....SUBROUTINE DATIN COMPUTES BY INTERPOLATION VALUES OF U, V, W,
C     ..... ETC.    BOTH FIRST AND SECOND ORDER INTERPOLATION CAN BE USED
C..........................................................................
      DIMENSION YY(32,10),Y0(10)
      COMMON/STRESS/TRR(61,32),TZZ(61,32),TTT(61,32),TRZ(61,32),
     1  TRT(61,32),TZT(61,32),EPSL(61,32)
      COMMON/VEL/U(61,41),V(61,41),W(61,41),P(61,41),DIV(61,41)
      COMMON/DAT/X(61),Y(61),R(61),Z(61),F(61),S(61),S2(61),C1(61)
     1,C2(61),C3(61),C4(61),C5(61),C6(61),C7(61),C10(61),C11(61),C12(61)
     2, TA,RE,H,K,N,M,AM,MM,MI2,HSI,KSI,HH,KH,E(61),F2(61),ZH(61),XH(61)
     3, AX,BX,AY,BY,NUMBER,NC,MC,EPS,NSTRT,GEF(61),TH(61),ITURB,NTURB
      COMMON/2/MPOINT,RT(32),UL(32),VL(32),WL(32),UU(32),VV(32),WW(32),
     1 WU(32),WV(32),UV(32),EPSIL(32),ISCHETZ,IDEP,IBLPRP
      NAMELIST/DAT7/MPOINT,RT,UL,VL,WL,UU,VV,WW,WU,WV,UV,EPSIL
      NAMELIST /DIS7/TSCALE,XKECON
      DATA TSCALE/.2/,XKECON/.53/
      DATA NMAX,IPT,NTAB,IORDER/32,-1,3,1/
C..........................................................................
C     .....THIS IS A TEMPORARY FUDGE
C     .....COMPUTE DISSIPATION RATE
      READ(5,DIS7)
      WRITE(6,DIS7)
      PRINT 11, NTURB,ISCHETZ,IDEP,IBLPRP
   11 FORMAT(//////////1H ,*NTURB,ISCHETZ,IDEP,IBLPRP = *,4I10)
      IF(IBLPRP.EQ.1) GO TO 60
C
      DO 50J=1,MPOINT
      XKEN = 0.5*(UU(J)+VV(J)+WW(J))
      EPSIL(J) = XKECON*SQRT(XKEN**3)/TSCALE
      IF(ISCHETZ.NE.0) UV(J)=0.
   50 CONTINUE
C..........................................................................
   60 CONTINUE
      IF(ITURB.EQ.1) CALL NOZERO
      PRINT 87
      WRITE(6,DAT7)
      IF(ITURB.EQ.1) NTAB=10
      DO 1J = 1,MPOINT
      YY(J,1) = UL(J)
      YY(J,2) = VL(J)
      YY(J,3) = WL(J)
      IF(ITURB.EQ.0) GO TO 1
      YY(J,4) = UU(J)
      YY(J,5) = VV(J)
      YY(J,6) = WW(J)
      YY(J,7) = WU(J)
      YY(J,8) = WV(J)
      YY(J,9) = UV(J)
      YY(J,10) = EPSIL(J)
    1 CONTINUE
      DO 2J = 1,N
      R0 = R(J)
      CALL IUNI(NMAX,MPOINT,RT,NTAB,YY,IORDER,R0,Y0,IPT,IERR)
      U(I,J) = Y0(1)
```

DATIN

```
      V(1,J) = Y0(2)
      W(1,J) = Y0(3)
      IF(ITURB.EQ.0) GO TO 2
      TRR(1,J) = Y0(4)
      TTT(1,J) = Y0(5)
      TZZ(1,J) = Y0(6)
      TRZ(1,J) = Y0(7)
      TZT(1,J) = Y0(8)
      TRT(1,J) = Y0(9)
      EPSL(1,J) = Y0(10)
    2 CONTINUE
      W(1,1) = (4.*W(1,2)-W(1,3))/3.
      IF(ITURB.EQ.0) GO TO 40
      TRR(1,1) = (4.*TRR(1,2)-TRR(1,3))/3.
      TZZ(1,1) = (4.*TZZ(1,2)-TZZ(1,3))/3.
      EPSL(1,1)=(4.*EPSL(1,2)-EPSL(1,3))/3.
      TTT(1,1)=(4.*TTT(1,2)-TTT(1,3))/3.
      IF(IDEP.EQ.0) GO TO 80
C   ...COMPUTE DISSIPATION RATE -- SET DISSIPATION EQUAL TO PRODUCTION
      DO 47 J=2,MM
      TEM1=TRR(1,J)*F(J)*(U(1,J+1)-U(1,J-1))
      TEM2=TTT(1,J)*U(1,J)*E(J)
      TEM3=TRT(1,J)*(F(J)*(V(1,J+1)-V(1,J-1))-V(1,J)*E(J))
      TEM4=TRZ(1,J)*F(J)*(W(1,J+1)-W(1,J-1))
   47 EPSL(1,J)= -(TEM1+TEM2+TEM3+TEM4)
      EMAX=ABS(EPSL(1,2))
      DO 65 J=3,MM
      TEPSL=ABS(EPSL(1,J))
      IF(TEPSL.LE.EMAX) GO TO 65
      EMAX=TEPSL
      JMAX=J
   65 CONTINUE
      JMAXM1=JMAX-1
      DO 70 J=1,JMAXM1
   70 EPSL(1,J)=EMAX
   80 CONTINUE
      IF(ISCHETZ.NE.1) GO TO 40
C     .....USE EDDY VISCOSITY TO COMPUTE TRT
      TRT(1,1) = 0.
      TRT(1,M) = 0.
      DO 42J = 2,MM
      DWR=(W(1,J+1)-W(1,J-1))
      IF(ABS(DWR) .LE. .01) GO TO 43
      DWR=F(J)*DWR
      XMU = TRZ(1,J)/DWR
      DVR = F(J)*(V(1,J+1)-V(1,J-1))
      VOR = V(1,J)*E(J)
      TRT(1,J) = XMU*(DVR-VOR)
      GO TO 42
   43 TRT(1,J) = 0.
   42 CONTINUE
C
   40 CONTINUE
      PRINT 90
      DO 5J = 1,M
      PRINT 91,J,R(J),U(1,J),V(1,J),W(1,J)
```

DATIN

```
    5 CONTINUE
      IF(ITURB.EQ.1) GC TO 10
      PRINT 95
      RETURN
   10 CONTINUE
      PRINT 93
      DO 6J = 1,N
      PRINT 91,J,TRR(1,J),TTT(1,J),TZZ(1,J),TRT(1,J),TRZ(1,J),TZT(1,J),
     1EPSL(1,J)
    6 CONTINUE
   87 FORMAT(1H1)
   90 FORMAT(1H1,* UPSTREAM CONDITIONS     J, R, U, V, W*//)
   91 FORMAT(1X,I5,7E16.5)
   93 FORMAT(1H1,*UPSTREAM CONDITIONS      J, TRR, TTT, TZZ, TRT, TRZ, TZ
     1T, EPSL*//)
   95 FORMAT(1H1,20(/),40X,*L A M I N A R   F L C W*)
      RETURN
      END
```

```
      SUBROUTINE IUNI(NMAX,N,X,NTAB,Y,IORDER,X0,Y0,IPT,IERR)
C*********************************************************************
C*
C*    PURPOSE:
C*                SUBROUTINE IUNI USES FIRST OR SECOND ORDER
C*                LAGRANGIAN INTERPOLATION TO ESTIMATE THE VALUES
C*                OF A SET OF A SET OF FUNCTIONS AT A POINT X0.  IUNI
C*                USES ONE INDEPENDENT VARIABLE TABLE AND A DEPENDENT
C*                VARIABLE TABLE FOR EACH FUNCTION TO BE EVALUATED.
C*                THE ROUTINE ACCEPTS THE INDEPENDENT VARIABLES SPACED
C*                AT EQUAL OR UNEQUAL INTERVALS.  EACH DEPENDENT
C*                VARIABLE TABLE MUST CONTAIN FUNCTION VALUES CORRES-
C*                PONDING TO EACH X(I) IN THE INDEPENDENT VARIABLE
C*                TABLE.  THE ESTIMATED VALUES ARE RETURNED IN THE Y0
C*                ARRAY WITH THE N-TH VALUE OF THE ARRAY HOLDING THE
C*                VALUE OF THE N-TH FUNCTION VALUE EVALUATED AT X0.
C*
C*        USE:
C*                CALL IUNI(NMAX,N,X,NTAB,Y,IORDER,X0,Y0,IPT,IERR)
C*
C* PARAMETERS:
C*
C*        NMAX    THE MAXIMUM NUMBER OF POINTS IN THE INDEPENDENT
C*                VARIABLE ARRAY.
C*
C*           N    THE ACTUAL NUMBER OF POINTS IN THE INDEPENDENT
C*                ARRAY,WHERE N .LE. NMAX.
C*
C*           X    A ONE-DIMENSIONAL ARRAY, DIMENSIONED (NMAX) IN THE
C*                CALLING PROGRAM, WHICH CONTAINS THE INDEPENDENT
C*                VARIABLES.  THESE VALUES MUST BE STRICTLY MONOTONIC.
C*
C*        NTAB    THE NUMBER OF DEPENDENT VARIABLE TABLES
C*
C*           Y    A TWO-DIMENSIONAL ARRAY DIMENSIONED (NMAX,NTAB) IN
C*                THE CALLING PROGRAM.  EACH COLUMN OF THE ARRAY
C*                CONTAINS A DEPENDENT VARIABLE TABLE
C*
C*      IORDER    INTERPOLATION PARAMETER SUPPLIED BY THE USER.
C*
C*                =0  ZERO ORDER INTERPOLATION: THE FIRST FUNCTION
C*                    VALUE IN EACH DEPENDENT VARIABLE TABLE IS
C*                    ASSIGNED TO THE CORRESPONDING MEMBER OF THE Y0
C*                    ARRAY.  THE FUNCTIONAL VALUE IS ESTIMATED TO
C*                    REMAIN CONSTANT AND EQUAL TO THE NEAREST KNOWN
C*                    FUNCTION VALUE.
C*
C*          X0    THE INPUT POINT AT WHICH INTERPOLATION WILL BE
C*                PERFORMED.
C*
C*          Y0    A ONE-DIMENSIONAL ARRAY DIMENSIONED (NTAB) IN THE
C*                CALLING PROGRAM.  UPON RETURN THE ARRAY CONTAINS THE
C*                ESTIMATED VALUE OF EACH FUNCTION AT X0.
C*
C*         IPT    ON THE FIRST CALL IPT MUST BE INITIALIZED TO -1 SO
C*                THAT MONOTONICITY WILL BE CHECKED. UPON LEAVING THE
```

```
IUNI
C*                    ROUTINE IPT EQUALS THE VALUE OF THE INDEX OF THE X      *
C*                    VALUE PRECEDING X0 UNLESS EXTRAPOLATION WAS             *
C*                    PERFORMED.  IN THAT CASE THE VALUE OF IPT IS            *
C*                    RETURNED AS:                                            *
C*                    =0  DENOTES X0 .LT. X(1) IF THE X ARRAY IS IN           *
C*                        INCREASING ORDER AND X(1) .GT. X0 IF THE X ARRAY    *
C*                        IS IN DECREASING ORDER.                             *
C*                    =N  DENOTES X0 .GT. X(N) IF THE X ARRAY IS IN           *
C*                        INCREASING ORDER AND X0 .LT. X(N) IF THE X ARRAY    *
C*                        IS IN DECREASING ORDER.                             *
C*                                                                            *
C*                    ON SUBSEQUENT CALLS, IPT IS USED AS A POINTER TO        *
C*                    BEGIN THE SEARCH FOR X0.                                *
C*                                                                            *
C*            IERR    ERROR PARAMETER GENERATED BY THE ROUTINE                *
C*                    =0  NORMAL RETURN                                       *
C*                    =J  THE J-TH ELEMENT OF THE X ARRAY IS OUT OF ORDER     *
C*                    =-1 ZERO ORDER INTERPOLATION PERFORMED BECAUSE          *
C*                        IORDER =0.                                          *
C*                    =-2 ZERO ORDER INTERPOLATION PERFORMED BECAUSE ONLY     *
C*                        ONE POINT WAS IN X ARRAY.                           *
C*                    =-3 NO INTERPOLATION WAS PERFORMED BECAUSE              *
C*                        INSUFFICIENT POINTS WERE SUPPLIED FOR SECOND        *
C*                        ORDER INTERPOLATION.                                *
C*                    =-4 EXTRAPOLATION WAS PERFORMED                         *
C*                                                                            *
C*                    UPON RETURN THE PARAMETER IERR SHOULD BE TESTED IN      *
C*                    THE CALLING PROGRAM.                                    *
C*                                                                            *
C*    REQUIRED ROUTINES                 NONE                                  *
C*                                                                            *
C*    SOURCE                            CMPB ROUTINE NTLUP MODIFIED           *
C*                                      BY COMPUTER SCIENCES CORPORATION      *
C*                                                                            *
C*    LANGUAGE                          FORTRAN                               *
C*                                                                            *
C*                                                                            *
C*    DATE RELEASED                     AUGUST 1,1973                         *
C*                                                                            *
C*    LATEST REVISION                   AUGUST 1,1973                         *
C*                                                                            *
C******************************************************************************
      DIMENSION X(1),Y(NMAX,1),Y0(1)
      NM1=N-1
      IERR=0
      J=1
C
C          TEST FOR ZERO ORDER INTERPOLATION
C
      IF (IORDER .EQ. 0) GO TO 10
      IF (N.LT. 2) GO TO 20
      GO TO 50
   10 IERR=-1
      GO TO 30
   20 IERR=-2
   30 DO 40 NT=1,NTAB
```

```
TUNI
            Y0(NT)=Y(1,NT)
   40       CONTINUE
         RETURN
   50    IF (IPT .GT. -1) GO TO 65
C
C                 CHECK FOR TABLE OF NODE POINTS BEING STRICTLY MONOTONIC
C                 THE SIGN OF DELX SIGNIFIES WHETHER TABLE IS IN
C                 INCREASING OR DECREASING ORDER.
C
         DELX=X(2)-X(1)
         IF (DELX .EQ. 0) GO TO 190
         IF (N .EQ. 2) GO TO 65
C
C                 CHECK FOR SIGN CONSISTENCY IN THE DIFFERENCES OF
C                 SUBSEQUENT PAIRS
C
         DO 60 J=2,NM1
            IF (DELX * (X(J+1)-X(J))) 190,190,60
   60       CONTINUE
C
C                 IPT IS INITIALIZED TO BE WITHIN THE INTERVAL
C
   65    IF (IPT .LT. 1) IPT=1
         IF (IPT .GT. NM1) IPT=NM1
         IN= SIGN (1.0,DELX *( X0-X(IPT)))
   70    P= X(IPT) - X0
         IF (P* (X(IPT +1)- X0)) 90,180,80
   80    IPT =IPT +IN
C
C                 TEST TO SEE IF IT IS NECCESARY TO EXTRAPOLATE
C
         IF (IPT.GT.0 .AND. IPT .LT. N) GO TO 70
         IERR=-4
         IPT=IPT- IN
C
C                 TEST FOR ORDER OF INTERPOLATION
C
C
   90    IF (IORDER .GT. 1) GO TO 120
C
C                 FIRST ORDER INTERPOLATION
C
         DO 100 NT=1,NTAB
             Y0(NT)=Y(IPT,NT)+((Y(IPT+1,NT)- Y(IPT,NT))*(X0-X(IPT)))/
        1            (X(IPT+1)-X(IPT))
  100       CONTINUE
         IF (IERR .EQ. -4) IPT=IPT+IN
         RETURN
C
C                 SECOND ORDER INTERPOLATION
C
  120    IF (N .EQ. 2) GO TO 200
C
C                 CHOOSING A THIRD POINT SO AS TO MINIMIZE THE DISTANCE
C                 BETWEEN THE THREE POINTS USED TO INTERPOLATE
C
```

```
IUNI
      IF (IPT .EQ. NM1) GO TO  140
      IF (IPT .EQ. 1) GO TO 130
      IF (DELX *(X0-X(IPT-1)).LT.DELX* (X(IPT+2)-X0)) GO TO 140
  130 L=IPT
      GO TO 150
  140 L=IPT -1
  150 V1=X(L)-X0
      V2=X(L+1)-X0
      V3=X(L+2)-X0
      DO 160 NT=1,NTAB
      YY1=(Y(L,NT) * V2 - Y(L+1,NT) * V1)/(X(L+1) - X(L))
      YY2=(Y(L+1,NT)*V3-Y(L+2,NT) *V2)/(X(L+2)-X(L+1))
  160 Y0(NT)=(YY1*V3 -YY2*V1)/(X(L+2)-X(L))
      IF (IERR .EQ. -4) IPT=IPT + IN
      RETURN
  180 IF(P .NE. 0) IPT=IPT +1
      DO 185 NT=1,NTAB
         Y0(NT)=Y(IPT,NT)
  185    CONTINUE
      RETURN
C
C              IERR IS SET TO THE SUBSCRIPT OF THE MEMBER OF THE TABLE
C              WHICH IS OUT OF ORDER
C
  190 IERR=J +1
      RETURN
  200 IERR=-3
      RETURN
      END
```

NOZERO

```
      SUBROUTINE NOZERO
      COMMON /2/MPOINT,RT(32),UL(32),VL(32),WL(32),UU(32),VV(32),WW(32),
     1 WU(32),WV(32),UV(32),EPSIL(32),ISCHETZ,IDEP,IRLPRP
      DIMENSION FIN(32),FOUT(32)
      EQUIVALENCE (FIN,FOUT)
C
      R2=RT(MPOINT)
      MPM1=MPOINT-1
      IG=0
C
      DO 11 IR=1,MPOINT
   11 FIN(IR)=UU(IR)
      GO TO 100
   15 CONTINUE
      DO 19 IR=1,MPOINT
   19 UU(IR)=FOUT(IR)
C
      DO 21 IR=1,MPOINT
   21 FIN(IR)=VV(IR)
      GO TO 100
   25 CONTINUE
      DO 29 IR=1,MPOINT
   29 VV(IR)=FOUT(IR)
C
      DO 31 IR=1,MPOINT
   31 FIN(IR)=WW(IR)
      GO TO 100
   35 CONTINUE
      DO 39 IR=1,MPOINT
   39 WW(IR)=FOUT(IR)
C
      DO 41 IR=1,MPOINT
   41 FIN(IR)=EPSIL(IR)
      GO TO 100
   45 CONTINUE
      DO 49 IR=1,MPOINT
   49 EPSIL(IR)=FOUT(IR)
```

```
C
      DO 59 IR=LIRP1,MPM1
      TRBINT=SQRT(UU(IR)**2 + VV(IR)**2 + WW(IR)**2)
      EPSHI=TRBINT * 1.E-4
      IF(EPSIL(IR) .GT. EPSHI) EPSIL(IR)=EPSHI
   59 CONTINUE
```

```
C
      RETURN
C
C  FIND LAST NON-ZERO VALUE
  100 CONTINUE
      LIR=MPOINT
      DO 110 IR=2,MPOINT
      IF(ABS(FIN(IR)) .GT. 1.E-8) GO TO 110
      LIR=IR-1
      GO TO 120
  110 CONTINUE
C
  120 CONTINUE
      IF(LIR .GE. MPM1) GO TO 190
C
C  DO LINEAR INTERPOLATION
      A=FIN(LIR)/(RT(LIR)-R2)
      LIRP1=LIR+1
      DO 150 IR=LIRP1,MPM1
```

```
NOZERO
              FOUT(IR)=A*(RT(IR)-R2)
          150 CONTINUE
        C
          190 CONTINUE
              IG=IG+1
              GO TO (15,25,35,45), IG
        C
              END
```

STRESS2

```
      SUBROUTINE STRESS2(IOFF)
C..........................................................................
C     .....SUBROUTINE STRESS2 COMPUTES REYNOLDS STRESSES USING THE MODEL
C     .....OF HANJALIC AND LAUNDER WITH THE PRESSURE-MEAN STRAIN
C     .....CORRELATION OF LAUNDER, REECE, AND RODI.
C     .....
C     .....NML=0 FOR DALY-HARLOW TURBULENT DIFFUSION
C     .....NML=1 FOR HANJALIC-LAUNDER TURBULENT DIFFUSION
C..........................................................................
      REAL KEC,KEN,KES,KOEC,KOEN,KOES,MUC,MUN,MUS,MURTN,MURTS,MURZN,
     1 MURZS
      REAL K,KS,KSI,KI,KH,K2I
      DIMENSION CP(32),CM(32),FE(32),AEP(32),BEP(32),CEP(32),ARR(32),
     1 BRR(32),CRR(32),ATT(32),BTT(32),CTT(32),AZZ(32),BZZ(32),CZZ(32),
     2 ART(32),BRT(32),CRT(32),ARZ(32),BRZ(32),CRZ(32),AZT(32),BZT(32),
     3 CZT(32),EEP(32),ERR(32),ETT(32),EZZ(32),ERT(32),ERZ(32),EZT(32),
     4 DEP(32),DRR(32),DTT(32),DZZ(32),DRT(32),DRZ(32),DZT(32)
      COMMON/2/MPOINT,RT(32),UL(32),VL(32),WL(32),UU(32),VV(32),WW(32),
     1 WU(32),WV(32),UV(32),EPSIL(32),ISCHETZ,IDEP,IALPRP
      COMMON/VISC/ARTVIS(61)
      COMMON/VLIF/RP(41),JND(41)
      COMMON/VEL/U(61,41),V(61,41),W(61,41),P(61,41),DIV(61,41)
      COMMON/DAT/X(61),Y(61),R(61),Z(61),F(61),S(61),S2(61),C1(61)
     1,C2(61),C3(61),C4(61),C5(61),C6(61),C7(61),C10(61),C11(61),C12(61)
     2, TA,RE,H,K,N,M,NM,MM,MI2,HSI,KSI,HH,KH,E(61),F2(61),ZH(61),XH(61)
     3, AX,BX,AY,BY,NUMBER,NC,MC,EPS,NSTRT,GEF(61),TH(61),ITURB,MTURB
      COMMON/PRESCO/ AM(61),AN(61),BM(61),BN(61),CX(61),CN(61),EW(61),
     1  EMS(61),FH(61),FHS(61),RH(61),SH(61),SHS(61),THNM,WORK(600),
     2  BCO1,BCO2,BCO3,SI,TI,SIS,SIREI,MM2,IFLG,H2I,K2I
      COMMON/STRESS/TRR(61,32),TZZ(61,32),TTT(61,32),TRZ(61,32),
     1  TRT(61,32),TZT(61,32),EPSL(61,32)
      NAMELIST/DATA/NML,CEPS,CEPS1,CEPS2,CSN,CS0,CS1,CON1,CON2,ISRCTT
      DATA NCALL/0/
      DATA CEPS,CEPS1,CEPS2,CS0,CS1,CON1,CON2/0.15,1.44,1.90,0.25,0.11,
     1  1.5,0.4/
      DATA ISRCTT/1/
      IF(NCALL.EQ.1) GO TO 10
      NCALL=1
      DO 11J = 2,MM
      CP(J) = 4.*F(J)*E(J)*RH(J)*FH(J)
   11 CM(J) = 4.*F(J)*E(J)*RH(J-1)*FH(J-1)
      READ(5,DATA)
      CSN=CS0
      IF(NML.EQ.1)CSN=CS1
      XNML=FLOAT(NML)
      XNMP1 = FLOAT(NML+1)
      X2NMP1=FLOAT(2*NML+1)
      X3NMP1 = FLOAT(3*NML+1)
      TAI2 = 2./TA
      CSN2 = 2.*CSN
      DO 12J = 1,MM
   12 FE(J) = F(J)*E(J)
      CON3 = (CON2+8.)/11.
      CON4 = (30.*CON2-2.)/55.
      CON5 = (8.*CON2-2.)/11.
      PRINT 900
```

STRESS2

```
      IF(NML.EQ.0) PRINT 901
      IF(NML.EQ.1) PRINT 902
      IF(IOFF.EQ.1) PRINT 903
      WRITE(6,DAT8)
      EEP(1) = 1.
      ERR(1) = 1.
      ETT(1) = 1.
      EZZ(1) = 1.
      ERT(1) = 0.
      ERZ(1) = 0.
      EZT(1) = 0.
      DEP(1) = 0.
      DRR(1) = 0.
      DTT(1) = 0.
      DZZ(1) = 0.
      DRT(1) = 0.
      DRZ(1) = 0.
      DZT(1) = 0.
   10 CONTINUE
      MSAVE=M
      MMSAVE=MM
      M=MTURB
      MM=MTURB-1
      UU(M) = 0.
      VV(M) = 0.
      WW(M) = 0.
      WU(M) = 0.
      WV(M) = 0.
      UV(M) = 0.
      EPSIL(M) = 0.
      DO 50 J=M,MSAVE
      DO 50 I=1,N
      TRR(I,J)=0.
      TTT(I,J)=0.
      TZZ(I,J)=0.
      TRT(I,J)=0.
      TRZ(I,J)=0.
      TZT(I,J)=0.
   50 EPSL(I,J)=0.
      IF(IDEP.EQ.0) GO TO 80
C  ...COMPUTE DISSIPATION RATE
      DO 47 J=2,MM
      TEM1=TRR(1,J)*F(J)*(U(1,J+1)-U(1,J-1))
      TEM2=TTT(1,J)*U(1,J)*F(J)
      TEM3=TRT(1,J)*(F(J)*(V(1,J)-V(1,J-1))-V(1,J)*E(J))
      TEM4=TRZ(1,J)*F(J)*(W(1,J+1)-W(1,J-1))
      TEM5=TZZ(1,J)*S(1)*(-3.*W(1,J)+4.*W(2,J)-W(3,J))
      TEM6=TRZ(1,J)*S(1)*(-3.*U(1,J)+4.*U(2,J)-U(3,J))
      TEM7=TZT(1,J)*S(1)*(-3.*V(1,J)+4.*V(2,J)-V(3,J))
   47 EPSL(1,J)= -(TEM1+TEM2+TFM3+TEM4+TEM5+TEM6+TEM7)
      EMAX=ABS(EPSL(1,2))
      DO 65 J=3,MM
      TEPSL=ABS(EPSL(1,J))
      IF(TEPSL.LE.FMAX) GO TO 65
      EMAX=TEPSL
      JMAX=J
```

STRESS2

```
   65 CONTINUE
      JMAXM1=JMAX-1
      DO 70 J=1,JMAXM1
   70 EPSL(1,J)=EMAX
   80 CONTINUE
      DO 1000I = 2,N
C     .....FUDGE TO KEEP THINGS WELL BEHAVED
      EPSL(I,M) = 1.0E+99
      DO 20J = 2,MM
      LC = I+JND(J)
      LN = I+JND(J+1)
      LS = I+JND(J-1)
C     .....COMPUTE ELEMENTS OF THE TRIDI MATRICES
      KEC = 0.5*(TRR(LC)+TTT(LC)+TZZ(LC))
      KEN = 0.5*(TRR(LN)+TTT(LN)+TZZ(LN))
      KES = 0.5*(TRR(LS)+TTT(LS)+TZZ(LS))
      MUC = KEC*TRP(LC)/EPSL(LC)
      MUN = KEN*TRP(LN)/EPSL(LN)
      MUS = KES*TRP(LS)/EPSL(LS)
      TJMH = 0.5*CM(J)*(MUS+MUC)
      TJPH = 0.5*CP(J)*(MUN+MUC)
      FU = F(J)*U(LC)
      AEP(J) = -FU-CEPS*TJMH
      CEP(J) = FU-CEPS*TJPH
      BEP(J) = TAI2-(AEP(J)+CEP(J))
      TJMH = CSN*TJMH
      TJPH = CSN*TJPH
      ARR(J) = -FU-X2NHP1*TJMH
      CRR(J) = FU-X2NHP1*TJPH
      BRR(J) = TAI2-(ARR(J)+CRR(J))
      ATT(J) = -FU-TJMH
      CTT(J) = FU-TJPH
      BTT(J) = TAI2-(ATT(J)+CTT(J))
      AZZ(J) = ATT(J)
      BZZ(J) = BTT(J)
      CZZ(J) = CTT(J)
      ART(J) = -FU-XNHP1*TJMH
      CRT(J) = FU-XNHP1*TJPH
      BRT(J) = TAI2-(ART(J)+CRT(J))
      ARZ(J) = ART(J)
      BRZ(J) = BRT(J)
      CRZ(J) = CRT(J)
      AZT(J) = ATT(J)
      BZT(J) = BTT(J)
      CZT(J) = CTT(J)
      IF(NHL.EQ.0) GO TO 20
      TEM = CSN*MUC*FE(J)
      ATT(J) = ATT(J)+2.*TEM
      CTT(J) = CTT(J)-2.*TEM
      ART(J) = ART(J)+2.*TEM
      CRT(J) = CRT(J)-2.*TEM
      AZT(J) = AZT(J)+TEM
      CZT(J) = CZT(J)-TEM
   20 CONTINUE
C     .....COMPUTE NON-HOMOGENEOUS TERMS
      DO 200J = 2,MM
```

STRESS2

```
      LC = I+JND(J)
      LN = I+JND(J+1)
      LS = I+JND(J-1)
      EJS = E(J)*E(J)
C     .....COMPUTE MEAN VELOCITY DERIVATIVES
      DUR = F(J)*(U(LN)-U(LS))
      DVR = F(J)*(V(LN)-V(LS))
      DWR = F(J)*(W(LN)-W(LS))
      UOR = U(LC)*F(J)
      VOR = V(LC)*F(J)
      IF(I.EQ.N) GO TO 100
      DVZ = S(I)*(V(LC+1)-V(LC-1))
      DUZ = S(I)*(U(LC+1)-U(LC-1))
      DWZ = S(I)*(W(LC+1)-W(LC-1))
      GO TO 101
  100 DUZ = S2(I)*(U(LC)-U(LC-1))
      DVZ = S2(I)*(V(LC)-V(LC-1))
      DWZ = S2(I)*(W(LC)-W(LC-1))
  101 CONTINUE
C     .....COMPUTE TURBULENT KINETIC ENERGIES
      KEC = 0.5*(TRR(LC)+TTT(LC)+TZZ(LC))
      KEN = 0.5*(TRR(LN)+TTT(LN)+TZZ(LN))
      KES = 0.5*(TRR(LS)+TTT(LS)+TZZ(LS))
      KOEC = KEC/EPSL(LC)
      KOEN = KEN/EPSL(LN)
      KOES = KES/EPSL(LS)
C     .....COMPUTE PRODUCTION RATE TERMS
      XKE23 = 2.*KEC/3.
      EPSOK=1./KOEC
      CEPS1OK = CEPS1*EPSOK
      TEM1 = TRR(LC)*DUR+TZZ(LC)*DWZ+TTT(LC)*UOR+TRT(LC)*(DVR-VOR)
     1 +TRZ(LC)*(DUZ+DWR)+TZT(LC)*DVZ
      PKE = -TEM1
      PEP = -CEPS1OK*TEM1
      PRR = -2.*(TRR(LC)*DUR+TRZ(LC)*DUZ-TRT(LC)*VOR)
      PTT = -2.*(TRT(LC)*DVR+TZT(LC)*DVZ+TTT(LC)*UOR)
      PZZ = -2.*(TRZ(LC)*DWR+TZZ(LC)*DWZ)
      PRT = -(TRR(LC)*DVR+TRZ(LC)*DVZ+TRT(LC)*DUR+TZT(LC)*DUZ+TRT(LC)*
     1 UOR-TTT(LC)*VOR)
      PRZ = -(TRR(LC)*DWR+TRZ(LC)*(DWZ+DUR)+TZZ(LC)*DUZ-TZT(LC)*VOR)
      PZT = -(TRZ(LC)*DVR+TZZ(LC)*DVZ+TRT(LC)*DWR+TZT(LC)*(DWZ+UOR))
C     .....COMPUTE D (I,J) TERMS IN LAUNDER#S PRESSURE-MEAN STRAIN
C     .....CORRELATION    (USE Q INSTEAD OF D)
      QRR = -2.*(TRR(LC)*DUR+TRZ(LC)*DWR+TRT(LC)*DVR)
      QTT = -2.*(TTT(LC)*UOR-TRT(LC)*VOR)
      QZZ = -2.*(TRZ(LC)*DUZ+TZT(LC)*DVZ+TZZ(LC)*DWZ)
      QRT = -(TTT(LC)*DVR+TZT(LC)*DWR+TRT(LC)*(DUR+UOR)-TRR(LC)*VOR)
      QRZ = -(TRR(LC)*DUZ+TZZ(LC)*DWR+TRZ(LC)*(DUR+DWZ)+TRT(LC)*DVZ
     1 +TZT(LC)*DVR)
      QZT = -(TZT(LC)*(DWZ+UOR)+TTT(LC)*DVZ+TRT(LC)*DUZ-TRZ(LC)*VOR)
C     .....COMPUTE PRESSURE-FLUCTUATING STRAIN CORRELATION TERM
C     .....    (ROTTA#S TERM)
      C1EPSOK = CON1*EPSOK
      PHIRR = -C1EPSOK*(TRR(LC)-XKE23)
      PHITT = -C1EPSOK*(TTT(LC)-XKE23)
      PHIZZ = -C1EPSOK*(TZZ(LC)-XKE23)
```

STRESS2

```
      PHIRT = -C1EPSOK*TRT(LC)
      PHIRZ = -C1EPSOK*TRZ(LC)
      PHIZT = -C1EPSOK*TZT(LC)
C     ....COMPUTE PRESSURE-MEAN STRAIN CORRELATION TERM
C     .....   (LAUNDER#S TERM)
      PKE23 = 2.*PKE/3.
      C4KE = CON4*KEC
      PH2RR = -CON3*(PRR-PKE23)-2.*C4KE*DUR-CON5*(QRR-PKE23)
      PH2TT = -CON3*(PTT-PKE23)-2.*C4KE*UOR-CON5*(QTT-PKE23)
      PH2ZZ = -CON3*(PZZ-PKE23)-2.*C4KE*DWZ-CON5*(QZZ-PKE23)
      PH2RT = -CON3*PRT-C4KE*(DVR-VOR)-CON5*QRT
      PH2RZ = -CON3*PRZ-C4KE*(DWR+DUZ)-CON5*QRZ
      PH2ZT = -CON3*PZT-C4KE*DVZ-CON5*QZT
C     .....COMPUTE NORMAL STRESS DISSIPATION RATE
      DIS = -2.*EPSL(LC)/3.
C     ....COMPUTE DISSIPATION RATE OF DISSIPATION RATE
      DISEPS = -CEPS2*EPSL(LC)*EPSOK
C     .....COMPUTE RHS TURBULENT DIFFUSION TERMS
      TEM1 = -CSN2*FE(J)*(KOEN*TRT(LN)**2-KOES*TRT(LS)**2)
      TEM2 = TRT(LC)*F(J)*(TRT(LN)-TRT(LS))+TTT(LC)*(TRR(LC)-TTT(LC))
     1 *E(J)
      TEM2 = -CSN2*XNHP1*E(J)*KOEC*TEM2
      DIFRR = X2NHP1*TEM1+TEM2
      DIFTT = -TEM1-TEM2
      DIFZZ = 0.
      TEM1 = KOEN*TRT(LN)*(TRR(LN)-TTT(LN))
      TEM2 = KOES*TRT(LS)*(TRR(LS)-TTT(LS))
      TEM3 = XNHP1*CSN*FE(J)*(TEM1-TEM2)
      TEM4 = F(J)*TRT(LC)*((TRR(LN)-TRR(LS))-X2NHP1*(TTT(LN)-TTT(LS)))
      TEM5 = 4.*XNHP1*TTT(LC)*TRT(LC)*E(J)
      DIFRT = TEM3+CSN*E(J)*KOEC*(TEM4-TEM5)
      TEM1 = FE(J)*(KOEN*TRT(LN)*TZT(LN)-KOES*TRT(LS)*TZT(LS))
      TEM2 = KOEC*(FE(J)*TRT(LC)*(TZT(LN)-TZT(LS))+TTT(LC)*TRZ(LC)*EJS)
      DIFRZ = -CSN*(X3NHP1*TEM1+XNHP1*TEM2)
      TEM1 = FE(J)*(KOEN*TRT(LN)*TRZ(LN)-KOES*TRT(LS)*TRZ(LS))
      TEM2 = KOEC*(FE(J)*TRT(LC)*(TRZ(LN)-TRZ(LS))-TTT(LC)*TZT(LC)*EJS)
      DIFZT = CSN*(TEM1+TEM2)
      IF(NML.EQ.0) GO TO 102
C     .....ADDITIONAL TERMS
      TEM3 = F(J)*TRR(LC)*(TTT(LN)-TTT(LS))+2.*E(J)*TRT(LC)*TRT(LC)
      TEM4 = CSN2*E(J)*KOEC*TEM3
      DIFRR = DIFRR-TEM4
      MURTN = 0.5*(TRT(LN)*KOEN+TRT(LC)*KOEC)
      MURTS = 0.5*(TRT(LC)*KOEC+TRT(LS)*KOES)
      TEM1 = CP(J)*MURTN*(TRT(LN)-TRT(LC))
      TEM2 = CM(J)*MURTS*(TRT(LC)-TRT(LS))
      TEM3 = FE(J)*(KOEN*TTT(LN)*(TRR(LN)-TTT(LN))-KOES*TTT(LS)*(TRR(LS)
     1 -TTT(LS)))
      DIFTT = DIFTT+TEM4+CSN2*(TEM1-TEM2+TEM3)
      MURZN = 0.5*(TRZ(LN)*KOEN+TRZ(LC)*KOEC)
      MURZS = 0.5*(TRZ(LC)*KOEC+TRZ(LS)*KOES)
      TEM1 = CP(J)*MURZN*(TRZ(LN)-TRZ(LC))
      TEM2 = CM(J)*MURZS*(TRZ(LC)-TRZ(LS))
      TEM3 = -FE(J)*(KOEN*TZT(LN)**2-KOES*TZT(LS)**2)
      DIFZZ = CSN2*(TEM1-TEM2+TEM3)
      TEM1 = CP(J)*MURTN*(TRR(LN)-TRR(LC))
```

STRESS2

```
      TEM2 = CM(J)*MURTS*(TRR(LC)-TRR(LS))
      TEM3 = -2.*FF(J)*(KOEN*TRT(LN)*TTT(LN)-KOES*TRT(LS)*TTT(LS))
      TEM4 = -2.*FE(J)*KOEC*TRT(LC)*(TTT(LN)-TTT(LS))
      DIFRT = DIFRT+CSN*(TEM1-TEM2+TEM3+TEM4)
      TEM1 = CP(J)*MURZN*(TRR(LN)-TRR(LC))
      TEM2 = CM(J)*MURZS*(TRR(LC)-TRR(LS))
      TEM3 = -KOEC*(FE(J)*TRZ(LC)*(TTT(LN)-TTT(LS))+2.*TZT(LC)*TRT(LC)
     1 *EJS)
      DIFRZ = DIFRZ+CSN*(TEM1-TEM2+TEM3)
      TEM1 = MURZN*(TRT(LN)-TRT(LC))
      TEM2 = MURTN*(TRZ(LN)-TRZ(LC))
      TEM3 = MURZS*(TRT(LC)-TRT(LS))
      TEM4 = MURTS*(TRZ(LC)-TRZ(LS))
      TEM5 = -FE(J)*(KOEN*TTT(LN)*TZT(LN)-KOES*TTT(LS)*TZT(LS))
      TEM6 = KOEC*(FE(J)*TRZ(LC)*(TRT(LN)-TRT(LS))+TRT(LC)*TRZ(LC)*EJS)
      TEM7 = CP(J)*(TEM1+TEM2)-CM(J)*(TEM3+TEM4)
      DIFZT = DIFZT+CSN*(TEM7+TEM5+TEM6)
  102 CONTINUE
C     .....COMPUTE RHS CONVECTION TERMS
      IF(I.EQ.N) GO TO 110
      WABS = ABS(W(LC))
      WWW= W(LC)+ARTVIS(I)*WABS
      WC = 2.*ARTVIS(I)*WABS
      WE = W(LC)-ARTVIS(I)*WABS
      CONRR = -S(I)*(TRR(LC+1)*WE+TRR(LC)*WC-TRR(LC-1)*WWW)
      CONTT = -S(I)*(TTT(LC+1)*WE+TTT(LC)*WC-TTT(LC-1)*WWW)
      CONZZ = -S(I)*(TZZ(LC+1)*WE+TZZ(LC)*WC-TZZ(LC-1)*WWW)
      CONRT = -S(I)*(TRT(LC+1)*WE+TRT(LC)*WC-TRT(LC-1)*WWW)
      CONRZ = -S(I)*(TRZ(LC+1)*WE+TRZ(LC)*WC-TRZ(LC-1)*WWW)
      CONZT = -S(I)*(TZT(LC+1)*WE+TZT(LC)*WC-TZT(LC-1)*WWW)
      CONEP = -S(I)*(EPSL(LC+1)*WE+EPSL(LC)*WC-EPSL(LC-1)*WWW)
      GO TO 111
  110 CONTINUE
      S2W = S2(I)*W(LC)
      CONRR = -S2W*(TRR(LC)-TRR(LC-1))
      CONTT = -S2W*(TTT(LC)-TTT(LC-1))
      CONZZ = -S2W*(TZZ(LC)-TZZ(LC-1))
      CONRT = -S2W*(TRT(LC)-TRT(LC-1))
      CONRZ = -S2W*(TRZ(LC)-TRZ(LC-1))
      CONZT = -S2W*(TZT(LC)-TZT(LC-1))
      CONEP = -S2W*(EPSL(LC)-EPSL(LC-1))
  111 CONTINUE
      CONRR = TAI2*TRR(LC)+2.*VOR*TRT(LC)+CONRR
      CONTT = TAI2*TTT(LC)-2.*VOR*TRT(LC)+CONTT
      CONZZ = TAI2*TZZ(LC)+CONZZ
      CONRT = TAI2*TRT(LC)+VOR*(TTT(LC)-TRR(LC))+CONRT
      CONRZ = TAI2*TRZ(LC)+VOR*TZT(LC)+CONRZ
      CONZT = TAI2*TZT(LC)-VOR*TRZ(LC)+CONZT
      CONEP = TAI2*EPSL(LC)+CONEP
C     .....ASSEMBLE NON-HOMOGENEOUS TERMS
      DEP(J) = CONEP+DISEPS+PEP
      DRR(J) = CONRR+PRR+DIS+PH1RR+PH2RR+DIFRR
      DTT(J) = CONTT+PTT+DIS+PH1TT+PH2TT+DIFTT
      DZZ(J) = CONZZ+PZZ+DIS+PH1ZZ+PH2ZZ+DIFZZ
      DRT(J) = CONRT+PRT+PH1RT+PH2RT+DIFRT
      DRZ(J) = CONRZ+PRZ+PH1RZ+PH2RZ+DIFRZ
```

STRESS2

```
      DZT(J) = CONZT+PZT+PH1ZT+PH2ZT+DIFZT
  200 CONTINUE
C     .....APPLY NUEMAN BOUNDARY CONDITIONS TO SECOND ORDER
      B1EPP = 3.*CFP(2)-AEP(2)
      B1RRP = 3.*CRR(2)-ARR(2)
      B1ZZP = 3.*CZZ(2)-AZZ(2)
      C1EPP = -4.*CEP(2)-BEP(2)
      C1RRP = -4.*CRR(2)-BRR(2)
       C1ZZP = -4.*CZZ(2)-BZZ(2)
      D1EPP = -DEP(2)
      D1RRP = -DRR(2)
      D1ZZP = -DZZ(2)
      EEP(1) = -C1FPP/B1EPP
      ERR(1) = -C1RRP/B1RRP
      EZZ(1) = -C1ZZP/B1ZZP
      DEP(1) = D1EPP/B1EPP
      DRR(1) = D1RRP/B1RRP
      DZZ(1) = D1ZZP/B1ZZP
      IF(ISBCTT.EQ.0) GO TO 310
      B1TTP = 3.*CTT(2)-ATT(2)
      C1TTP = -4.*CTT(2)-BTT(2)
      D1TTP = -DTT(2)
      ETT(1) = -C1TTP/B1TTP
      DTT(1) = D1TTP/B1TTP
  310 CONTINUE
      DO 300J = 2,MM
      JM = J-1
      ZZ = 1./(BEP(J)+AEP(J)*EEP(JM))
      EEP(J) = -CEP(J)*ZZ
      DEP(J) = (DEP(J)-AEP(J)*DEP(JM))*ZZ
      ZZ = 1./(BRR(J)+ARR(J)*ERR(JM))
      ERR(J) = -CRR(J)*ZZ
      DRR(J) = (DRR(J)-ARR(J)*DRR(JM))*ZZ
      ZZ = 1./(BZZ(J)+AZZ(J)*EZZ(JM))
      EZZ(J) = -CZZ(J)*ZZ
      DZZ(J) = (DZZ(J)-AZZ(J)*DZZ(JM))*ZZ
      ZZ = 1./(BRT(J)+ART(J)*ERT(JM))
      ERT(J) = -CRT(J)*ZZ
      DRT(J) = (CRT(J)-ART(J)*DRT(JM))*ZZ
      ZZ = 1./(BRZ(J)+ARZ(J)*ERZ(JM))
      ERZ(J) = -CRZ(J)*ZZ
      DRZ(J) = (CRZ(J)-ARZ(J)*DRZ(JM))*ZZ
      ZZ = 1./(BZT(J)+AZT(J)*EZT(JM))
      EZT(J) = -CZT(J)*ZZ
      DZT(J) = (DZT(J)-AZT(J)*DZT(JM))*ZZ
      IF(ISBCTT.EQ.0) GO TO 300
      ZZ = 1./(BTT(J)+ATT(J)*ETT(JM))
      ETT(J) = -CTT(J)*ZZ
      DTT(J) = (CTT(J)-ATT(J)*DTT(JM))*ZZ
  300 CONTINUE
      IF(I.EQ.2) GO TO 302
      DO 303J = 1,M
      LW = I-1+JND(J)
      TRR(LW) = UU(J)
      TTT(LW) = VV(J)
      TZZ(LW) = WW(J)
```

STRESS2

```
      TRT(LW) = UV(J)
      TZT(LW) = WV(J)
      TRZ(LW) = WU(J)
      EPSL(LW) = EPSIL(J)
  303 CONTINUE
  302 CONTINUE
      DO 301J = 1,MM
      L = M-J
      LP = L+1
      EPSIL(L) = DEP(L)+EEP(L)*EPSIL(LP)
      UU(L) = DRR(L)+ERR(L)*UU(LP)
      WW(L) = DZZ(L)+EZZ(L)*WW(LP)
      UV(L) = DRT(L)+ERT(L)*UV(LP)
      WU(L) = DRZ(L)+ERZ(L)*WU(LP)
      WV(L) = DZT(L)+EZT(L)*WV(LP)
      IF(ISBCTT.EQ.0) GO TO 301
      VV(L) = DTT(L)+ETT(L)*VV(LP)
  301 CONTINUE
      IF(ISBCTT.NE.0) GO TO 1000
      ETT(1) = 0.
      DTT(1) = UU(1)
      DO 305J = 2,MM
      ZZ = 1./(BTT(J)+ATT(J)*ETT(J-1))
      ETT(J) = -CTT(J)*ZZ
  305 DTT(J) = (DTT(J)-ATT(J)*DTT(J-1))*ZZ
      DO 306J = 1,MM
      L = M-J
  306 VV(L) = DTT(L)+ETT(L)*VV(L+1)
 1000 CONTINUE
      DO 304J = 1,M
      LW = N+JND(J)
      EPSL(LW) = EPSIL(J)
      TRR(LW) = UU(J)
      TTT(LW) = VV(J)
      TZZ(LW) = WW(J)
      TRT(LW) = UV(J)
      TRZ(LW) = WU(J)
      TZT(LW) = WV(J)
  304 CONTINUE
      M=MSAVE
      MM=MMSAVE
  900 FORMAT(1H1,*H A N J A L I C   L A U N D E R   S E C O N D   O R D
     1E R   C L O S U R E*//1X,*T U R B U L E N C E   M O D E L*////1X,*
     2LAUNDER-REECE-RODI PRESSURE-MEAN STRAIN CORRELATION*//)
  901 FORMAT(1H0,//* D A L Y - H A R L O W   T U R B U L E N T   D I F F
     1U S I O N*)
  902 FORMAT(1H0,//* H A N J A L I C - L A U N D E R   T U R B U L E N T
     1   D I F F U S I O N*////)
  903 FORMAT(////,1X,* IN THIS CALCULATION TURBULENCE IS UNCOUPLED FROM
     1THE MEAN FLOW*/////)
      RETURN
      END
```

PRESSR

```
      SUBROUTINE PRESSR(IOFF)
      REAL K,KS,KSI,KI,KM,K2I
      COMMON/VLIF/RP(41),JND(41)
      COMMON/VEL/U(61,41),V(61,41),W(61,41),P(61,41),DIV(61,41)
      COMMON/DAT/X(61),Y(61),R(61),Z(61),F(61),S(61),S2(61),C1(61)
     1,C2(61),C3(61),C4(61),C5(61),C6(61),C7(61),C10(61),C11(61),C12(61)
     2, TA,RE,H,K,N,M,AM,MM,MI2,HSI,KSI,HH,KH,E(61),F2(61),PH(61),XH(61)
     3, AX,BX,AY,BY,NUMBER,NC,MC,EPS,NSTRT,GEF(61),TH(61),ITURB,NTURB
      COMMON/PRESCO/ AM(61),AN(61),BM(61),BN(61),CM(61),CN(61),EM(61),
     1  EMS(61),FM(61),FMS(61),RM(61),SM(61),SMS(61),THNM,WORK(600),
     2  BCO1,BCO2,BCO3,SI,TI,SIS,SIREI,MM2,IFLG,K21,K2I
      COMMON/UPSTRM/WH(61),WHRR(61),WI,VI,ALPH,RMAX,ZMAX,REI,THETAI,RR
      COMMON/STRESS/TRR(61,32),TZZ(61,32),TTT(61,32),TRZ(61,32),
     1  TRT(61,32),TZT(61,32),EPSL(61,32)
      COMMON/TRIDAT2/AU(61),AV(61),AW(61),BU(61),BV(61),BW(61),CU(61),
     1  CV(61),CW(61),EU(61),EV(61),EW(61),FE(32),FE2(32)
      COMMON/VISC/ARTVIS(61)
      DIMENSION CXE(61),CXW(61),CYE(32),CYW(32),FFI(32),SSI(61)
      IF(NCALL.EQ.1) GC TO 9
      NCALL=1
      PRINT 91
      DO 4I=1,N
    4 SSI(I) = 1./(S(I)*2.*H)
      DO 5J = 1,M
    5 FFI(J) = 1./(F(J)*2.*K)
    8 NN2 = N-2
      TAI2 = 2./TA
      MM2 = M-2
    9 CONTINUE
C.......................................................................
C     .....REVISED PRESSURE CALCULATION
      DO 100I = 1,N
      IF(I.GT.1) GO TO 103
      DWX = -3.*W(1,1)+4.*W(2,1)-W(3,1)
      CXE(1) = -+2I*W(1,1)*DWX
      CYE(1) = 0.
      DO 101J = 2,MM
      DWY = W(1,J+1)-W(1,J-1)
      DWX = -3.*W(1,J)+4.*W(2,J)-W(3,J)
      CXE(J) = -SSI(1)*(F(J)*U(1,J)*DWY+S(1)*W(1,J)*DWX)
      DUY = U(1,J+1)-U(1,J-1)
      DUX = -3.*U(1,J)+4.*U(2,J)-U(3,J)
  101 CYE(J) = -FFI(J)*(F(J)*U(1,J)*DUY+S(1)*W(1,J)*DUX-V(1,J)**2*E(J))
      DWY = W(1,MM2)-4.*W(1,MM)+3.*W(1,M)
      CXE(M) = -SSI(1)*(F(M)*U(1,M)*DWY)
      DUY = U(1,MM2)-4.*U(1,MM)+3.*U(1,M)
      DUX = -3.*U(1,M)+4.*U(2,M)-U(3,M)
      CYE(M) = -FFI(M)*(F(M)*U(1,M)*DUY+S(1)*W(1,M)*DUX-V(1,M)**2*F(M))
      GO TO 100
  103 IF(I.EQ.N) GO TO 120
      CXW(I) = CXE(I)
      CYW(I) = CYE(I)
      LC = I+JND(I)
      ARTABW = ARTVIS(I)*ABS(W(LC))
      TEM1 = W(LC)-ARTABW
      TEM3 = W(LC)+ARTABW
```

PRESSR

```
      CXE(1) = -+2I*(W(LC+1)*TEM1+2.*W(LC)*ARTABW-W(LC-1)*TEM3)
      CYE(1) = 0.
      DO 102J = 2,MM
      LN = I+JND(J+1)
      LC = I+JND(J)
      LS = I+JND(J-1)
      CXW(J) = CXE(J)
      CYW(J) = CYE(J)
      TEM1 = F(J)*U(LC)*(W(LN)-W(LS))
      ARTABW = ARTVIS(1)*ABS(W(LC))
      TEM2 = W(LC)-ARTABW
      TEM3 = 2.*ARTABW
      TEM4 = W(LC)+ARTABW
      CXE(J) = -SSI(I)*(TEM1+S(I)*(W(LC+1)*TEM2+W(LC)*TEM3-W(LC-1)*TEM4)
     1)
      TEM1 = F(J)*U(LC)*(U(LN)-U(LS))
  102 CYE(J) = -FFI(J)*(TEM1+S(I)*(U(LC+1)*TEM2+U(LC)*TEM3-U(LC-1)*TEM4)
     1 -V(LC)**2*E(J))
      CXW(M) = CXE(M)
      CYW(M) = CYE(M)
      LC = I+JND(M)
      DWY = W(I,MM2)-4.*W(I,MM)+3.*W(LC)
      CXE(M) = -SSI(I)*F(M)*U(LC)*DWY
      DUY = U(I,MM2)-4.*U(I,MM)+3.*U(I,M)
      ARTABW = ARTVIS(1)*ABS(W(LC))
      TEM2 = W(LC)-ARTABW
      TEM3 = 2.*ARTABW
      TEM4 = W(LC)+ARTABW
      CYE(M) = -FFI(M)*(F(M)*U(LC)*DUY+S(I)*(U(LC+1)*TEM2+U(LC)*TEM3-
     1 U(LC-1)*TEM4)-V(LC)**2*E(M))
      GO TO 110
  120 CONTINUE
      CXW(1) = CXE(1)
      CYW(1) = CYE(1)
      CXE(1) = -SSI(N)*(S2(N)*W(N,1)*(W(N+1)-W(NP,1)))
      CYE(1) = 0.
      DO 121J = 2,MM
      CXW(J) = CXE(J)
      CYW(J) = CYE(J)
      LN = N+JND(J+1)
      LS = N+JND(J-1)
      LC = N+JND(J)
      CXE(J) = -SSI(N)*(S2( N)*W(LC)*(W(LC)-W(LC-1))+F(J)*U(LC)*(W(LN)-
     1 W(LS)))
      TEM1 = F(J)*U(LC)*(U(LN)-U(LS))
      TEM2 = S2(N)*W(LC)*(U(LC)-U(LC-1))
  121 CYE(J) = -FFI(J)*(TEM1+TEM2-V(LC)**2*E(J))
      CXW(M) = CXE(M)
      CYW(M) = CYE(M)
      TEM1 = F(M)*U(N,M)*(W(N,MM2)-4.*W(N,MM)+3.*W(N,M))
      TEM3 = S2(N)*W(N,M)*(U(N,M)-U(NM,M))
      TEM2 = F(M)*U(N,M)*(U(N,MM2)-4.*U(N,MM)+3.*U(N,M))
      CXE(M) = -SSI(N)*TEM1
      CYE(M) = -FFI(M)*(TEM2+TEM3-V(N,M)**2*E(M))
  110 CONTINUE
C     .....COMPUTE RHS OF PRESSURE EQUATION
```

```
PRESSR
            DO 111J = 1,MM
            LC = I+JND(J)
            DCYY = CYE(J+1)+CYW(J+1)-CYE(J)-CYW(J)
            CYAV = 0.25*(CYE(J+1)+CYW(J+1)+CYE(J)+CYW(J))
            DCXX = CXE(J+1)-CXW(J+1)+CXE(J)-CXW(J)
            CXAV = 0.25*(CXE(J+1)+CXW(J+1)+CXE(J)+CXW(J))
            TEM1 = 2.*K*FHS(J)*DCYY
            TEM2 = K*GEF(J)*CYAV
            TEM3 = 2.*H*SHS(I-1)*DCXX
            TEM4 = H*TH(I-1)*CXAV
        111 P(LC-1) = TEM1+TEM2+TEM3+TEM4 +TAI2*DIV(LC-1)
        100 CONTINUE
C          MODIFY P(I,J) TO INCLUDE PRESSURE BOUNDARY CONDITIONS
C        B.C. AT X = 0
            DO 11J = 1,MM
            L = I+JND(J)
            LP = I+JND(J+1)
            WXJ = H2I*(-3.*W(L)+4.*W(L+1)-W(L+2))
            WXJP = H2I*(-3.*W(LP)+4.*W(LP+1)-W(LP+2))
            WXXJ = HSI*(2.*W(L)-5.*W(L+1)+4.*W(L+2)-W(L+3))
            WXXJP = HSI*(2.*W(LP)-5.*W(LP+1)+4.*W(LP+2)-W(LP+3))
            WX = 0.5*(WXJ+WXJP)
            WXX = 0.5*(WXXJ+WXXJP)
            TEM1 = WHRR(J)
            TEM2 = SIS*WXX+T1*WX
            TEM3 = SIREI*(TEM1+TEM2)
            DWY = W(1,J+1)-W(1,J)
            UHF = 0.5*(U(1,J+1)+U(1,J))
            TEM3 = TEM3-2.*FH(J)*UHF*DWY/S1
            TEM4 = -WH(J)*WX+TEM3
         11 P(L) = P(L)+TEM4*BCO1
C          EAST BOUDARY CONDITION
            DO 12J = 1,MM
         12 P(NM,J) = P(NM,J)-P(N,J)*(16.*SHS(NM)+2.*TH(NM))
C        B.C. AT Y = 0.5
            FM=2.*F(M)*K
            DO 13I = 1,NM
            TEM1 = (V(I,M)+V(I+1,M))**2+(U(I,M)+U(I+1,M))**2
            TEM1 = TEM1*0.25
            TEM1 = TEM1/R(M)
            TEM2 = SH(I)*(W(I,M)+W(I+1,M))*(U(I+1,M)-U(I,M))
         13 P(I,MM) = P(I,MM)+BCO3*(TEM1-TEM2)/FM
            IF(ITURB.EQ.0) GO TO 24
            IF(IOFF.EQ.1) GO TO 24
C     .....MODIFY NONHOMOGENEOUS TERMS IN PRESSURE EQUATION FOR REYNOLDS
C     .....STRESSES
            DO 300I = 1,N
            IF(I.GT.1) GO TO 303
            DZZX = -3.*TZZ(1,1)+4.*TZZ(2,1)-TZZ(3,1)
            DRZY = -3.*TRZ(1,1)+4.*TRZ(1,2)-TRZ(1,3)
            CXE(1) = -SSI(1)*(S(1)*DZZX+2.*F(1)*DRZY)
            CYE(1) = 0.
            DO 301J = 2,MM
            DZZX = -3.*TZZ(1,J)+4.*TZZ(2,J)-TZZ(3,J)
```

PRESSR

```
      DRZY = TRZ(1,J+1)-TRZ(1,J-1)
      CXE(J) = -SSI(1)*(S(1)*DZZX+F(J)*DRZY+TRZ(1,J)*E(J))
      DRZX = -3.*TRZ(1,J)+4.*TRZ(2,J)-TRZ(3,J)
      DRRY = TRR(1,J+1)-TRR(1,J-1)
  301 CYE(J) = -FFI(J)*(S(1)*DRZX+F(J)*DRRY+(TRR(1,J)-TTT(1,J))*E(J))
      DRZX = -3.*TRZ(1,M)+4.*TRZ(2,M)-TRZ(3,M)
      DRZY = TRZ(1,MM2)-4.*TRZ(1,MM)+3.*TRZ(1,M)
      DRRY = TRR(1,MM2)-4.*TRR(1,MM)+3.*TRR(1,M)
      DZZX = -3.*TZZ(1,M)+4.*TZZ(2,M)-TZZ(3,M)
      CXE(M) = -SSI(1)*(S(1)*DZZX+F(M)*DRZY+TRZ(1,M)*E(M))
      CYE(M) = -FFI(M)*(S(1)*DRZX+F(M)*DRRY+(TRR(1,M)-TTT(1,M))*E(M))
      GO TO 300
  303 IF(I.EQ.N) GO TO 320
      CXW(1) = CXE(1)
      CYW(1) = CYE(1)
      LC = I+JND(1)
      DZZX = TZZ(LC+1)-TZZ(LC-1)
      DRZY = -3.*TRZ(I,1)+4.*TRZ(I,2)-TRZ(I,3)
      CYE(1) = 0.
      CXE(1) = -SSI(I)*(S(I)*DZZX+2.*F(1)*DRZY)
      DO 302J = 2,MM
      LN = I+JND(J+1)
      LC = I+JND(J)
      LS = I+JND(J-1)
      CXW(J) = CXE(J)
      CYW(J) = CYE(J)
      DRZX = TRZ(LC+1)-TRZ(LC-1)
      DRRY = TRR(LN)-TRR(LS)
      DZZX = TZZ(LC+1)-TZZ(LC-1)
      DRZY = TRZ(LN)-TRZ(LS)
      CXE(J) = -SSI(I)*(S(I)*DZZX+F(J)*DRZY+TRZ(LC)*E(J))
  302 CYE(J) = -FFI(J)*(S(I)*DRZX+F(J)*DRRY+(TRR(LC)-TTT(LC))*E(J))
      CXW(M) = CXE(M)
      CYW(M) = CYE(M)
      LC = I+JND(M)
      DRZX = TRZ(LC+1)-TRZ(LC-1)
      DRRY = TRR(I,MM2)-4.*TRR(I,MM)+3.*TRR(I,M)
      DZZX = TZZ(LC+1)-TZZ(LC-1)
      DRZY = TRZ(I,MM2)-4.*TRZ(I,MM)+3.*TRZ(I,M)
      CXE(M) = -SSI(I)*(S(I)*DZZX+F(M)*DRZY+TRZ(LC)*E(M))
      CYE(M) = -FFI(M)*(S(I)*DRZX+F(M)*DRRY+(TRR(LC)-TTT(LC))*E(M))
      GO TO 310
  320 CONTINUE
      CXW(1) = CXE(1)
      CYW(1) = CYE(1)
      DZZX = TZZ(N,1)-TZZ(NM,1)
      DRZY = -3.*TRZ(N,1)+4.*TRZ(N,2)-TRZ(N,3)
      CYE(1) = 0.
      CXE(1) = -SSI(N)*(S2(N)*DZZX+2.*F(1)*DRZY)
      DO 321J = 2,MM
      CXW(J) = CXE(J)
      CYW(J) = CYE(J)
      LN = N+JND(J+1)
      LC = N+JND(J)
      LS = I+JND(J-1)
      DRZX = TRZ(LC)-TRZ(LC-1)
```

PRESSR

```
      DRRY = TRR(LN)-TRR(LS)
      DZZX = TZZ(LC)-TZZ(LC-1)
      DRZY = TRZ(LN)-TRZ(LS)
      CXE(J) = -SS1(N)*(S2(N)*DZZX+F(J)*DRZY+E(J)*TRZ(LC))
  321 CYE(J) = -FF1(J)*(S2(N)*DRZX+F(J)*DRRY+(TRR(LC)-TTT(LC))*E(J))
      CXW(M) = CXE(M)
      CYW(M) = CYE(M)
      LC = I+JND(M)
      DZZX = TZZ(LC)-TZZ(LC-1)
      DRZY = TRZ(N,MM2)-4.*TRZ(N,MM)+3.*TRZ(N,M)
      DRZX = TRZ(LC)-TRZ(LC-1)
      DRRY = TRR(N,MM2)-4.*TRR(N,MM)+3.*TRR(N,M)
      CXE(M) = -SS1(N)*(S2(N)*DZZX+F(M)*DRZY+E(M)*TRZ(LC))
      CYE(M) = -FF1(M)*(S2(N)*DRZX+F(M)*DRRY+(TRR(LC)-TTT(LC))*E(M))
  310 CONTINUE
C     .....COMPUTE RIGHT HAND SIDE OF PRESSURE EQUATION WITH TURBULENCE
      DO 311J = 1,MM
      LC = I+JND(J)
      DCYY = CYE(J+1)+CYW(J+1)-CYE(J)-CYW(J)
      CYAV = 0.25*(CYE(J+1)+CYW(J+1)+CYE(J)+CYW(J))
      DCXX = CXE(J+1)-CXW(J+1)+CXE(J)-CXW(J)
      CXAV = 0.25*(CXE(J+1)+CXW(J+1)+CXE(J)+CXW(J))
      TEM1 = 2.*K*FHS(J)*DCYY
      TEM2 = K*GEF(J)*CYAV
      TEM3 = 2.*H*SHS(I-1)*DCXX
      TEM4 = H*TH(I-1)*CXAV
      P(LC-1) = P(LC-1)+TEM1+TEM2+TEM3+TEM4
  311 CONTINUE
  300 CONTINUE
C     .....WEST BOUNDARY CONDITION INCLUDING REYNOLDS STRESS TERMS
      DO 25J = 1,MM
      LC = 1+JND(J)
      LN = 1+JND(J+1)
      TZZ1 = 0.5*(TZZ(LC)+TZZ(LN))
      TZZ2 = 0.5*(TZZ(LC+1)+TZZ(LN+1))
      TZZ3 = 0.5*(TZZ(LC+2)+TZZ(LN+2))
      TZZX1 = H21*(-3.*TZZ1+4.*TZZ2-TZZ3)
      RTRZY = R(J+1)*TRZ(LN)-R(J)*TRZ(LC)
      TEM1 = 2.*FH(J)*RTRZY/(RH(J)*S1)
   25 P(LC) = P(LC)-BC01*(TZZX1+TEM1)
C     .....NORTH BOUNDARY CONDITION
      DO 26I = 1,NM
      LS = I+JND(MM)
      TRRY = -TRR(LS)-TRR(LS+1)
   26 P(LS) = P(LS)-BC03*(F(M)*TRRY)/FM
   24 CONTINUE

   15 CALL BLKTRI(IFLG,1,MM,AM,BM,CM,1,NM,AN,BN,CN,61,P,IERROR,WORK)
      IF(IERROR.EQ.0) GO TO 14
      PRINT 90,IERROR
      STOP
   14 CONTINUE
      IF(IFLG.EQ.1) GO TO 16
      IFLG = 1
      GO TO 15
   16 CONTINUE
```

```
PRESSR
      90 FORMAT(1X,*IERROR =*,I5)                                    PRESSR.282
      91 FORMAT(1H1,20(/),20X,*REVISED PRESSURE CALCULATION, IPRCALC = 1*) PRESSR.283
         RETURN                                                      PRESSR.284
         END                                                         PRESSR.285
```

DIVCON

```
      SUBROUTINE DIVCON(DIV,TA,NT,NM,MM,CONCRIT,ICON)
C.......................................................................
C     .....SUBROUTINE DIVCON COMPUTES THE RMS DIVERGENCE WHICH IS
C     .....USED AS A CONVERGENCE CHECK
C.......................................................................
      DIMENSION DIV(61,41)
      DATA DIVOLD,DIVNEW,NTOLD/0.,0.,0/
      DIVOLD = DIVNEW
      DIVNEW = 0.
      DO 10I = 1,NM
      DO 10J = 1,MM
   10 DIVNEW = DIVNEW+DIV(I,J)*DIV(I,J)
      DIVNEW = SQRT(DIVNEW)
      DIVNEW = DIVNEW/FLOAT(NM*MM)
      DELDIV = DIVNEW-DIVOLD
      DELDIVT = DELDIV/(FLOAT(NT-NTOLD)*TA)
      DELDIVT = DELDIVT/(DIVNEW+DIVOLD)
      PRINT 90,NT,DIVNEW
      PRINT 91,NTOLD,DELDIVT
      NTOLD = NT
      IF(ABS(DELDIVT).LE.CONCRIT)ICON=1
      IF(ICON.EQ.1) PRINT 92,CONCRIT
   92 FORMAT(1H1,20(/),10X,*CONVERGENCE ACHIEVED WITH CONCRIT =*,E11.4/
   90 FORMAT(1H0,//,1X,*AFTER*,I5,3X,*TIME STEPS, RMS DIVERGENCE =*,E11
     1.4)
   91 FORMAT(1X,*BASED ON THIS AND ITS VALUE AFTER*,I5,3X,*STEPS, THE T
     1ME RATE OF CHANGE OF RMS DIV =*,E11.4)
      RETURN
      END
```

FLXCHK

```
      SUBROUTINE FLXCHK(U,W,R,S,F,TA,N,M,H,K,FLUXW,FLUXE,FLUXN)
C
C     .....SUBROUTINE FLXCHK COMPUTES MASS FLUX THROUGH WEST, NORTH AND
C     .....EAST BOUNDARIES.  FLUX OUT IS POSITIVE.
C
      REAL K
      DIMENSION U(61,41),W(61,41),R(61),S(61),F(61)
      FF=2.*K
      I = 1
   10 SUM = 0.
      TEM1 = W(I,1)*R(1)/F(1)
      TEM2 = 3.*W(I,2)*R(2)/F(2)
      TEM3 = 3.*W(I,3)*R(3)/F(3)
      TEM4 = W(I,4)*R(4)/F(4)
      SUM = 3.*K*(TEM1+TEM2+TEM3+TEM4)/(8.*FF)
      DO 20J = 4,M
      CC = 4.
      IF(MOD(J,2).EQ.0) CC=2.
      IF(J.EQ.4) CC=1.
      IF(J.EQ.M) CC=1.
      TEM1 = W(I,J)*R(J)/(F(J)*FF)
   20 SUM = SUM + K*CC*TEM1/3.
      IF(I.EQ.1) GO TO 21
      FLUXE = SUM
      GO TO 22
   21 FLUXW=-SUM
      I = N
      GO TO 10
   22 CONTINUE
      SS=2.*H
   30 SUM = 0.
      TEM1 = U(1,M)/S(1)
      TEM2 = 3.*U(2,M)/S(2)
      TEM3 = 3.*U(3,M)/S(3)
      TEM4 = U(4,M)/S(4)
      SUM = 3.*H*(TEM1+TEM2+TEM3+TEM4)/(8.*SS)
      DO 40I = 4,N
      CC = 4.0
      IF(MOD(I,2).EQ.0) CC=2.
      IF(I.EQ.4) CC=1.
      IF(I.EQ.N) CC=1.
      TEM1 = U(I,M)/(S(I)*SS)
   40 SUM = SUM+H*CC*TEM1/3.
      FLUXN = R(M)*SUM
      RETURN
      END
```

TERMCHK

```
      SUBROUTINE TERMCHK
C.......................................................................
C     .....SUBROUTINE TERMCHK COMPUTES AND COMPARES VARIOUS TERMS
C     .....IN THE EQUATIONS OF MOTION
C.......................................................................
      COMMON/VEL/U(61,41),V(61,41),W(61,41),P(61,41),DIV(61,41)
      COMMON/DAT/X(61),Y(61),R(61),Z(61),F(61),S(61),S2(61),C1(61)
     1,C2(61),C3(61),C4(61),C5(61),C6(61),C7(61),C10(61),C11(61),C12(61)
     2, TA,RE,M,K,N,M,AM,MM,M12,MSI,KSI,MH,KM,E(61),F2(61),ZH(61),XH(61)
     3, AX,BX,AY,BY,NUMBER,NC,MC,EPS,NSTRT,GEF(61),TH(61),ITURB,MTURB
      COMMON/STRESS/TRR(61,32),TZZ(61,32),TTT(61,32),TRZ(61,32),
     1  TRT(61,32),TZT(61,32),EPSL(61,32)
C     .....COMPARE THE AXIAL PRESSURE GRADIENT WITH THE REYNOLDS STRESSES
C     .....IN THE AXIAL MOMENTUM EQUATION AT THE AXIS
      J = 1
      PRINT 90,J
      DO 10I = 2,NM
      DZP = S2(I)*(P(I,1)-P(I-1,1))
      DZZZ = S(I)*(TZZ(I+1,1)-TZZ(I-1,1))
      DRZR = 2.*F(I)*(-3.*TRZ(I,1)+4.*TRZ(I,2)-TRZ(I,3))
      TURBW = DZZZ + DRZR
      PRINT 92,I,DZP,TURBW,DZZZ,DRZR
   10 CONTINUE
   90 FORMAT(1H1,*COMPARE PRESSURE GRADIENT AND REYNOLDS STRESS TERMS IN
     1 THE AXIAL MOMENTUM EQUATION*///5X,*J =*,I5,//,5X,*I,DZP,TURBW,DZ
     2Z,DRZR*//)
   92 FORMAT(3X,I5,5X,E11.4,5X,E11.4,5X,E11.4,5X,E11.4)
      RETURN
      END
```

BLKTRI

```
      SUBROUTINE BLKTRI (IFLG,NP,N,AN,BN,CN,MP,M,AM,BM,CM,IDIMY,Y,
     1                   IERROR,W)
C
C
C
C     SUBROUTINE BLKTRI SOLVES A SYSTEM OF LINEAR EQUATIONS OF THE FORM
C
C          AN(J)*X(I,J-1) + AM(I)*X(I-1,J) + (BN(J)+BM(I))*X(I,J)
C
C          + CN(J)*X(I,J+1) + CM(I)*X(I+1,J) = Y(I,J)
C
C               FOR I = 1,2,...,M  AND  J = 1,2,...,N.
C
C     I+1 AND I-1 ARE EVALUATED MODULO M AND J+1 AND J-1 MODULO N, I.E.,
C
C          X(I,0) = X(I,N),  X(I,N+1) = X(I,1),
C          X(0,J) = X(M,J),  X(M+1,J) = X(1,J).
C
C     THESE EQUATIONS USUALLY RESULT FROM THE DISCRETIZATION OF
C     SEPARABLE ELLIPTIC EQUATIONS.  BOUNDARY CONDITIONS MAY BE
C     DIRICHLET, NEUMANN, OR PERIODIC.
C
C
C     * * * * * * * * * *     ON INPUT     * * * * * * * * * *
C
C     IFLG
C       = 0  INITIALIZATION ONLY.  CERTAIN QUANTITIES THAT DEPEND ON NP,
C                                  N, AN, BN, AND CN ARE COMPUTED AND
C                                  STORED IN THE WORK ARRAY  W.
C       = 1  THE QUANTITIES THAT WERE COMPUTED IN THE INITIALIZATION ARE
C            USED TO OBTAIN THE SOLUTION X(I,J).
C
C       NOTE:  A CALL WITH IFLG = 0 TAKES APPROXIMATELY TWICE AS MUCH
C              TIME AS A CALL WITH IFLG = 1  .  HOWEVER, THE
C              INITIALIZATION DOES NOT HAVE TO BE REPEATED UNLESS NP, N,
C              AN, BN, OR CN CHANGE.
C
C     NP
C       = 0  IF AN(1) AND CN(N) ARE NOT ZERO, WHICH CORRESPONDS TO
C            PERIODIC BOUNARY CONDITIONS.
C       = 1  IF AN(1) AND CN(N) ARE ZERO.
C
C     N
C       THE NUMBER OF UNKNOWNS IN THE J-DIRECTION.  IF NP = 1, N MUST BE
C       OF THE FORM 2**K-1 WHERE K IS AN INTEGER .GT. 1  .  IF NP = 0, N
C       MUST BE OF THE FORM 2**K.  (THE OPERATION COUNT OF THE ALGORITHM
C       IS PROPORTIONAL TO MN LOG2N AND, HENCE, N SHOULD BE SELECTED
C       LESS THAN OR EQUAL TO M.)
C
C     AN,BN,CN
C       ONE-DIMENSIONAL ARRAYS OF LENGTH N THAT SPECIFY THE COEFFICIENTS
C       IN THE LINEAR EQUATIONS GIVEN ABOVE.
C
C     MP
C       = 0  IF AM(1) AND CM(M) ARE NOT ZERO, WHICH CORRESPONDS TO
C            PERIODIC BOUNDARY CONDITIONS.
```

```
BLKTRI
      C          = 1  IF AM(1) = CM(M) = 0  .
      C
      C        M
      C          THE NUMBER OF UNKNOWNS IN THE I-DIRECTION.  M MAY BE ANY INTEGER
      C          GREATER THAN 1  .
      C
      C        AM,BM,CM
      C          ONE-DIMENSIONAL ARRAYS OF LENGTH M THAT SPECIFY THE COEFFICIENTS
      C          IN THE LINEAR EQUATIONS GIVEN ABOVE.
      C
      C        IDIMY
      C          THE ROW (OR FIRST) DIMENSION OF THE TWO-DIMENSIONAL ARRAY Y AS
      C          IT APPEARS IN THE PROGRAM CALLING BLKTRI.  THIS PARAMETER IS
      C          USED TO SPECIFY THE VARIABLE DIMENSION OF Y.  IDIMY MUST BE AT
      C          LEAST M.
      C
      C        Y
      C          A TWO-DIMENSIONAL ARRAY THAT SPECIFIES THE VALUES OF THE RIGHT
      C          SIDE OF THE LINEAR SYSTEM OF EQUATIONS GIVEN ABOVE.  Y MUST BE
      C          DIMENSIONED AT LEAST M*N.
      C
      C        W
      C          A ONE-DIMENSIONAL ARRAY THAT MUST BE PROVIDED BY THE USER FOR
      C          WORK SPACE.  IF NP = 0, THE LENGTH OF W MUST BE AT LEAST
      C                           (2NLOG2(N)+N+2*MAX(4N,6M)),
      C                    IF NP = 1, (2(N+1)(LOG2(N+1)-1)+2*MAX(2N,6M)).
      C
      C
      C        * * * * * * * * * *     ON OUTPUT     * * * * * * * * * *
      C
      C        Y
      C          CONTAINS THE SOLUTION X.
      C
      C        IERROR
      C          AN ERROR FLAG THAT INDICATES INVALID INPUT PARAMETERS.  EXCEPT
      C          FOR NUMBER ZERO, A SOLUTION IS NOT ATTEMPTED.
      C
      C          = 0  NO ERROR.
      C          = 1  M .LT. 2  .
      C          = 2  N IS NOT THE FORM 2**K-1 WHEN NP = 1  .
      C          = 3  N IS NOT OF THE FORM 2**K WHEN NP = 0  .
      C          = 4  BLKTRI FAILED WHILE COMPUTING RESULTS THAT DEPEND ON THE
      C               COEFFICIENT ARRAYS AN, BN, CN.  CHECK THESE ARRAYS.
      C          = 5  IDIMY.LT. M.
      C
      C        W
      C          CONTAINS INTERMEDIATE VALUES THAT MUST NOT BE DESTROYED IF
      C          BLKTRI WILL BE CALLED AGAIN WITH IFLG = 1  .
      C
      C
      C
      C
            DIMENSION       AN(1)      ,BN(1)      ,CN(1)      ,AM(1)      ,
           1                BM(1)      ,CM(1)      ,Y(IDIMY,1) ,W(1)
            EXTERNAL        PROD       ,PRODP      ,CPROD      ,CPRODP
            COMMON /CBLKT/  NPP        ,K          ,EPS        ,CNV        ,
           1                L          ,NCMPLX     ,IK         ,IZ
```

```
BLKTRI
      IF (IFLG) 114,101,114
C
C TEST M AND N FOR THE PROPER FORM
C
  101 IERROR = 1
      IF (M-2) 117,102,102
  102 NM = N
      NPP = NP
      IF (NPP) 103,104,103
  103 NM = NM+1
  104 IK = 2
      K = 0
  105 IK = IK+IK
      K = K+1
      IF (NM/IK) 106,106,105
  106 IF (NPP) 107,109,107
  107 IERROR = 2
      IWCN = 2*(K+1)*(K-1)-N+3
      IW1 = IWCN+N
      IWAM = IW1
      IWBM = IWAM+N
      IF (K-2) 117,108,108
  108 NCK = 2**K-1
      IF (N-NCK) 117,111,117
  109 IERROR = 3
      IWCN = 2*K*N+3
      IW1 = IWCN+N
      IWAM = IW1
      IWBM = IWAM+N+N
      IF (K-2) 117,110,110
  110 NCK = 2**K
      IF (N-NCK) 117,111,117
C
C DIVIDE W INTO SEVERAL SUB WORKING ARRAYS
C
  111 IERROR = 5
      IF (IDIMY-M) 117,112,112
  112 IERROR = 0
      IW2 = IW1+M
      IW3 = IW2+M
      IWD = IW3+M
      IWW = IWD+M
      IWU = IWW+M
      W(IWCN) = CN(N)
      I = IWCN
      DO 113 J=2,N
         I = I+1
         W(I) = CN(J-1)
  113 CONTINUE
C
C SUBROUTINE COMPB COMPUTES THE ROOTS OF THE B POLYNOMIALS
C
      CALL COMPB (N,IERROR,AN,BN,W(IWCN),W,W(IWAM),W(IWBM))
      GO TO 117
  114 IF (MP) 115,116,115
C
```

```
BLKTRI
      C SUBROUTINE BLKTRI SOLVES THE LINEAR SYSTEM
      C
        115 CALL BLKTRI (N,AN,BN,W(IWCN),M,AM,BM,CM,IDIMY,Y,W,W(IW1),W(IW2),
           1             W(IW3),W(IWD),W(IWW),W(IWU),PROD,CPROD)
            GO TO 117
        116 CALL BLKTRI (N,AN,BN,W(IWCN),M,AM,BM,CM,IDIMY,Y,W,W(IW1),W(IW2),
           1             W(IW3),W(IWD),W(IWW),W(IWU),PRODP,CPRODP)
        117 CONTINUE
            RETURN
            END
```

```
BLKTRI
      SUBROUTINE BLKTRI (N,AN,BN,CN,M,AM,BM,CM,IDIMY,Y,B,W1,W2,W3,WD,
     1                   WW,WU,PRDCT,CPRDCT)
C
C BLKTRI SOLVES THE LINEAR SYSTEM
C
C B  CONTAINS THE ROOTS OF ALL THE B POLYNOMIALS
C W1,W2,W3,WD,WW,WU  ARE ALL WORKING ARRAYS
C PRDCT  IS EITHER PRODP OR PROD DEPENDING ON WHETHER THE BOUNDARY
C CONDITIONS IN THE M DIRECTION ARE PERIODIC OR NOT
C CPRDCT IS EITHER CPRODP OR CPROD WHICH ARE THE COMPLEX VERSIONS
C OF PRODP AND PROD. THESE ARE CALLED IN THE EVENT THAT SOME
C OF THE ROOTS OF THE B SUB P POLYNOMIAL ARE COMPLEX
C
C
      DIMENSION       AN(1)       ,BN(1)       ,CN(1)       ,AM(1)       ,
     1                BM(1)       ,CM(1)       ,B(1)        ,W1(1)       ,
     2                W2(1)       ,W3(1)       ,WD(1)       ,WW(1)       ,
     3                WU(1)       ,Y(IDIMY,1)
      COMMON /CBLKT/  NPP         ,K           ,EPS         ,CNV         ,
     1                L           ,NCMPLX      ,IK          ,IZ
      NH = 2**K
      IH1 = NH+NH
C
C BEGIN REDUCTION PHASE
C
      KDO = K
      IF (NPP) 101,102,101
  101 KDO = K-1
  102 DO 108 L=1,KDO
         IR = L-1
         IZ = 2**IR
         ISGN = (-1)**IR
         MSGN = -ISGN
         IH2 = 2**(K-IR+1)
         LM = (IR-2)*IH1+IH2+IH2+1
         LZ = (IR-1)*IH1+IH2+1
         IIM = IZ-1
         IIZ = IIM+IIM+1
         JM1 = IIM+IIM+LM
         I1 = IZ+IZ
         CALL PRDCT (IIZ,B(LZ),IIM,B(LM),IIM,B(JM1),0,DUM,Y(1,IZ),W3,M,
     1               AM,BM,CM,WD,WW,WU,ISGN)
         IF = 2**(K-IR-1)
         DO 107 JJ=1,IF
            I = JJ*I1
            I6 = I
            I7 = I-IZ
            I9 = I+IZ
            J2 = JJ+JJ
            J4 = J2+J2
            JM1 = (J4-2)*IIM+LM
            JP1 = J4*IIM+LM
            JP2 = J2*IIZ+LZ
            JP3 = (J4+2)*IIM+LM
            IF (JJ-IF) 105,103,103
  103       IF (NPP) 107,104,107
```

```
RLKTR1
      104       JP1 = LM
                JP2 = LZ
                JP3 = IIM+IIM+LM
                I6 = 0
                I9 = IZ
      105       CALL PRDCT (IIM,B(JM1),0,DUM,0,DUM,IP,AN(I7+1),W3,W1,M,AM,
         1                  BM,CM,WD,WW,WU,MSGN)
                CALL PRDCT (II7,B(JP2),IIM,B(JP1),IIM,B(JP3),0,DUM,Y(1,I9),
         1                  W3,M,AM,BM,CM,WD,WW,WU,ISGN)
                CALL PRDCT (IIM,B(JP1),0,DUM,0,DUM,IZ,CN(I6+1),W3,W2,M,AM,
         1                  BM,CM,WD,WW,WU,MSGN)
                DO 106 J=1,M
                   Y(J,I) = W1(J)+W2(J)-Y(J,I)
      106       CONTINUE
      107    CONTINUE
      108 CONTINUE
          IF (NPP) 112,109,112
      109 J2 = 2*N*(K-1)+3
          J1 = 2*N*(K-2)+5
          IF (NCMPLX) 110,111,110
      110 CALL CPRDCT (N,B(J2),N-1,B(J1),0,DUM,0,DUM,Y(1,N),Y(1,N),M,AM,BM,
         1             CM,W1,W3,WW,MSGN)
          GO TO 112
      111 CALL PRDCT (N,B(J2),N-1,B(J1),0,DUM,0,DUM,Y(1,N),Y(1,N),M,AM,BM,
         1            CM,WD,WW,WU,MSGN)
C
C BEGIN BACK SUBSTITUTION PHASE
C
      112 DO 126 LL=1,K
             L = K-LL+1
             IR = L-1
             ISGN = (-1)**IR
             MSGN = -ISGN
             IZ = 2**IR
             IIM = IZ-1
             IIZ = IIM+IIM+1
             IM2 = 2**(K-IR+1)
             LM = (IR-2)*IF1+IM2+IM2+1
             LZ = (IR-1)*IF1+IM2+1
             IF = 2**(K-IR)-1
             DO 125 JJ=1,IF,2
                I = JJ*IZ
                I5 = I-IZ
                I6 = I+IZ
                I7 = I5
                J2 = JJ+JJ
                JM1 = (J2-2)*IIM+LM
                JZ = (JJ-1)*II7+LZ
                JP1 = J2*IIM+LM
                IF (JJ-1) 113,113,117
      113       IF (NPP) 115,114,115
      114       I7 = N
                GO TC 117
      115       DO 116 J=1,M
                   W1(J) = 0.
      116       CONTINUE
```

BLKTRI

```
          GO TO 118
117       CALL PRDCT (IIM,A(JM1),0,DUM,0,DUM,I2,AN(IS+1),Y(1,I7),W1,
     1                M,AM,BM,CM,WD,WW,WU,MSGN)
118       IF (JJ-IF) 122,119,119
119       IF (NPP) 120,122,120
120       DO 121 J=1,M
             W2(J) = 0.
121       CONTINUE
          GO TO 123
122       CALL PRDCT (IIM,B(JP1),0,DUM,0,DUM,I2,CN(I+1),Y(1,I6),W2,M,
     1                AM,BM,CM,WD,WW,WU,MSGN)
123       DO 124 J=1,M
             W1(J) = Y(J,I)-W1(J)-W2(J)
124       CONTINUE
          CALL PRDCT (II2,B(JZ),IIM,B(JM1),IIM,B(JP1),0,DUM,W1,
     1                Y(1,I),M,AM,BM,CM,WD,WW,WU,ISGN)
125    CONTINUE
126 CONTINUE
    RETURN
    END
```

PROD

```
      SUBROUTINE PROD (ND,BD,NM1,BM1,NM2,BM2,NA,AA,X,Y,M,A,B,C,D,W,U,IS)
C
C PROD APPLIES A SEQUENCE OF MATRIX OPERATIONS TO THE VECTOR X AND
C STORES THE RESULT IN Y
C BD,BM1,BM2 ARE ARRAYS CONTAINING ROOTS OF CERTIAN B POLYNOMIALS
C ND,NM1,NM2 ARE THE LENGTHS OF THE ARRAYS BD,BM1,BM2 RESPECTIVELY
C AA   ARRAY CONTAINING SCALAR MULTIPLIERS OF THE VECTOR X
C NA IS THE LENGTH OF THE ARRAY AA
C X,Y  THE MATRIX OPERATIONS ARE APPLIED TO X AND THE RESULT IS Y
C A,B,C  ARE ARRAYS WHICH CONTAIN THE TRIDIAGONAL MATRIX
C M  IS THE ORDER OF THE MATRIX
C D,W,U ARE WORKING ARRAYS
C IS  DETERMINES WHETHER OR NOT A CHANGE IN SIGN IS MADE
C
      DIMENSION       A(1)       ,B(1)       ,C(1)       ,X(1)       ,
     1                Y(1)       ,D(1)       ,W(1)       ,BD(1)      ,
     2                BM1(1)     ,BM2(1)     ,AA(1)      ,U(1)
      IF (ND) 102,102,101
  101 IF (IS) 104,104,102
  102 DO 103 J=1,M
         W(J) = X(J)
         Y(J) = X(J)
  103 CONTINUE
      GO TO 106
  104 DO 105 J=1,M
         W(J) = -X(J)
         Y(J) = W(J)
  105 CONTINUE
  106 MM = M-1
      ID = ND
      IBR = 0
      M1 = NM1
      M2 = NM2
      IA = NA
  107 IF (IA) 110,110,108
  108 RT = AA(IA)
      IA = IA-1
C
C SCALAR MULTIPLICATION
C
      DO 109 J=1,M
         Y(J) = RT*W(J)
  109 CONTINUE
  110 IF (ID) 130,130,111
  111 RT = BD(ID)
      ID = ID-1
      IF (ID .EQ. 0) IBR = 1
C
C BEGIN SOLUTION TO SYSTEM
C
      D(M) = A(M)/(B(M)-RT)
      W(M) = Y(M)/(B(M)-RT)
      DO 112 J=2,MM
         K = M-J
         DEN = B(K+1)-RT-C(K+1)*D(K+2)
         D(K+1) = A(K+1)/DEN
```

PROD

```
      W(K+1) = (Y(K+1)-C(K+1)*W(K+2))/DEN
  112 CONTINUE
      DEN = B(1)-RT-C(1)*D(2)
      W(1) = 1.
      IF (DEN) 113,114,113
  113 W(1) = (Y(1)-C(1)*W(2))/DEN
  114 DO 115 J=2,M
        W(J) = W(J)-D(J)*W(J-1)
  115 CONTINUE
      IF (NA) 118,118,107
  116 DO 117 J=1,M
        Y(J) = W(J)
  117 CONTINUE
      IBR = 1
      GO TO 107
  118 IF (M1) 119,119,120
  119 IF (M2) 116,116,125
  120 IF (M2) 122,122,121
  121 IF (ABS(BM1(M1))-ABS(BM2(M2))) 125,125,122
  122 IF (IBR) 123,123,124
  123 IF (ABS(BM1(M1)-BD(ID))-ABS(BM1(M1)-RT)) 116,124,124
  124 RT = RT-BM1(M1)
      M1 = M1-1
      GO TO 128
  125 IF (IBR) 126,126,127
  126 IF (ABS(BM2(M2)-BD(ID))-ABS(BM2(M2)-RT)) 116,127,127
  127 RT = RT-BM2(M2)
      M2 = M2-1
  128 DO 129 J=1,M
        Y(J) = Y(J)+RT*W(J)
  129 CONTINUE
      GO TO 107
  130 RETURN
      END
```

PROOP

```
      SUBROUTINE PROOP (ND,RD,NM1,RM1,NM2,RM2,NA,AA,X,Y,M,A,B,C,D,U,W,
     1                  IS)
C
C PROOP APPLIES A SEQUENCE OF MATRIX OPERATIONS TO THE VECTOR X AND
C STORES THE RESULT IN Y         PERIODIC BOUNDARY CONDITIONS
C
C BD,BM1,BM2 ARE ARRAYS CONTAINING ROOTS OF CERTIAN B POLYNOMIALS
C ND,NM1,NM2 ARE THE LENGTHS OF THE ARRAYS BD,BM1,BM2 RESPECTIVELY
C AA   ARRAY CONTAINING SCALAR MULTIPLIERS OF THE VECTOR X
C NA IS THE LENGTH OF THE ARRAY AA
C X,Y  THE MATRIX OPERATIONS ARE APPLIED TO X ANC THE RESULT IS Y
C A,B,C  ARE ARRAYS WHICH CONTAIN THE TRIDIAGONAL MATRIX
C M  IS THE ORDER OF THE MATRIX
C D,U,W ARE WORKING ARRAYS
C IS  CETERMINES WHETHER OR NOT A CHANGE IN SIGN IS MADE
C
      DIMENSION       A(1)        ,B(1)        ,C(1)        ,X(1)        ,
     1                Y(1)        ,D(1)        ,U(1)        ,BD(1)       ,
     2                BM1(1)      ,BM2(1)      ,AA(1)       ,W(1)
      IF (ND) 102,102,101
  101 IF (IS) 104,104,102
  102 DO 103 J=1,M
         Y(J) = X(J)
         W(J) = X(J)
  103 CONTINUE
      GO TO 106
  104 DO 105 J=1,M
         Y(J) = -X(J)
         W(J) = Y(J)
  105 CONTINUE
  106 MM = M-1
      MM2 = M-2
      ID = ND
      IBR = 0
      M1 = NM1
      M2 = NM2
      IA = NA
  107 IF (IA) 110,110,108
  108 RT = AA(IA)
      IA = IA-1
      DO 109 J=1,M
         Y(J) = RT*W(J)
  109 CONTINUE
  110 IF (ID) 131,131,111
  111 RT = BD(ID)
      ID = ID-1
      IF (ID .EQ. 0) IBR = 1
C
C BEGIN SOLUTION TO SYSTEM
C
      BM = B(M)-RT
      YM = Y(M)
      DEN = B(1)-RT
      D(1) = C(1)/DEN
      U(1) = A(1)/DEN
      W(1) = Y(1)/DEN
```

```
PRODP
          V = C(M)
          DO 112 J=2,MM2
             DEN = B(J)-RT-A(J)*D(J-1)
             D(J) = C(J)/DEN
             U(J) = -A(J)*U(J-1)/DEN
             W(J) = (Y(J)-A(J)*W(J-1))/DEN
             BM = BM-V*U(J-1)
             YM = YM-V*W(J-1)
             V = -V*C(J-1)
      112 CONTINUE
          DEN = B(M-1)-RT-A(M-1)*D(M-2)
          D(M-1) = (C(M-1)-A(M-1)*U(M-2))/DEN
          W(M-1) = (Y(M-1)-A(M-1)*W(M-2))/DEN
          AM = A(M)-V*D(M-2)
          BM = BM-V*U(M-2)
          YM = YM-V*W(M-2)
          DEN = BM-AM*D(M-1)
          IF (DEN) 113,114,113
      113 W(M) = (YM-AM*W(M-1))/DEN
          GO TO 115
      114 W(M) = 1.
      115 W(M-1) = W(M-1)-D(M-1)*W(M)
          DO 116 J=2,MM
             K = M-J
             W(K) = W(K)-D(K)*W(K+1)-U(K)*W(M)
      116 CONTINUE
          IF (NA) 119,119,107
      117 DO 118 J=1,M
             Y(J) = W(J)
      118 CONTINUE
          IBR = 1
          GO TO 107
      119 IF (M1) 120,120,121
      120 IF (M2) 117,117,126
      121 IF (M2) 123,123,122
      122 IF (ABS(BM1(M1))-ABS(BM2(M2))) 126,126,123
      123 IF (IBR) 124,124,125
      124 IF (ABS(BM1(M1)-ED(ID))-ABS(BM1(M1)-RT)) 117,125,125
      125 RT = RT-BM1(M1)
          M1 = M1-1
          GO TO 129
      126 IF (IBR) 127,127,128
      127 IF (ABS(BM2(M2)-ED(ID))-ABS(BM2(M2)-RT)) 117,128,128
      128 RT = RT-BM2(M2)
          M2 = M2-1
      129 DO 130 J=1,M
             Y(J) = Y(J)+RT*W(J)
      130 CONTINUE
          GO TO 107
      131 RETURN
          END
```

```
CPROD
      SUBROUTINE CPROD (ND,BD,NM1,BM1,AM2,BM2,NA,AA,X,YY,M,A,B,C,D,W,Y,
     1                  ISGN)
C
C PROD APPLIES A SEQUENCE OF MATRIX OPERATIONS TO THE VECTOR X AND
C STORES THE RESULT IN YY              (COMPLEX CASE)
C AA    ARRAY CONTAINING SCALAR MULTIPLIERS OF THE VECTOR X
C ND,NM1,NM2 ARE THE LENGTHS OF THE ARRAYS BD,BM1,BM2 RESPECTIVELY
C BD,BM1,BM2 ARE ARRAYS CONTAINING ROOTS OF CERTIAN B POLYNOMIALS
C NA IS THE LENGTH OF THE ARRAY AA
C X,YY THE MATRIX OPERATIONS ARE APPLIED TO X AND THE RESULT IS YY
C A,B,C  ARE ARRAYS WHICH CONTAIN THE TRIDIAGONAL MATRIX
C M  IS THE ORDER OF THE MATRIX
C D,W,Y ARE WORKING ARRAYS
C ISGN  DETERMINES WHETHER OR NOT A CHANGE IN SIGN IS MADE
C
      DIMENSION       A(1)       ,B(1)       ,C(1)       ,X(1)       ,
     1                Y(1)       ,D(1)       ,W(1)       ,BD(1)      ,
     2                BM1(1)     ,BM2(1)     ,AA(1)      ,YY(1)
      COMPLEX         Y          ,D          ,W          ,BD         ,
     1                CRT        ,DEN        ,Y1         ,Y2
      IF (ND) 102,102,101
  101 IF (ISGN) 104,104,102
  102 DO 103 J=1,M
         Y(J) = CMPLX(X(J),0.)
  103 CONTINUE
      GO TO 106
  104 DO 105 J=1,M
         Y(J) = CMPLX(-X(J),0.)
  105 CONTINUE
  106 MM = M-1
      ID = ND
      M1 = NM1
      M2 = NM2
      IA = NA
  107 IFLG = 0
      IF (ID) 114,114,108
  108 CRT = BD(ID)
      ID = ID-1
      IFLG = 1
C
C BEGIN SOLUTION TO SYSTEM
C
      D(M) = A(M)/(B(M)-CRT)
      W(M) = Y(M)/(B(M)-CRT)
      DO 109 J=2,MM
         K = M-J
         DEN = B(K+1)-CRT-C(K+1)*D(K+2)
         D(K+1) = A(K+1)/DEN
         W(K+1) = (Y(K+1)-C(K+1)*W(K+2))/DEN
  109 CONTINUE
      DEN = B(1)-CRT-C(1)*D(2)
      IF (CABS(DEN)) 110,111,110
  110 Y(1) = (Y(1)-C(1)*W(2))/DEN
      GO TO 112
  111 Y(1) = (1.,0.)
  112 DO 113 J=2,M
```

CPROD

```
      Y(J) = W(J)-D(J)*Y(J-1)
  113 CONTINUE
  114 IF (M1) 115,115,117
  115 IF (M2) 126,126,116
  116 RT = BM2(M2)
      M2 = M2-1
      GO TO 122
  117 IF (M2) 118,118,119
  118 RT = BM1(M1)
      M1 = M1-1
      GO TO 122
  119 IF (ABS(BM1(M1))-ABS(BM2(M2))) 121,121,120
  120 RT = BM1(M1)
      M1 = M1-1
      GO TO 122
  121 RT = BM2(M2)
      M2 = M2-1
  122 Y1 = (B(1)-RT)*Y(1)+C(1)*Y(2)
      IF (MM-2) 125,123,123
C
C MATRIX MULTIPLICATION
C
  123 DO 124 J=2,MM
         Y2 = A(J)*Y(J-1)+(B(J)-RT)*Y(J)+C(J)*Y(J+1)
         Y(J-1) = Y1
         Y1 = Y2
  124 CONTINUE
  125 Y(M) = A(M)*Y(M-1)+(B(M)-RT)*Y(M)
      Y(M-1) = Y1
      IFLG = 1
      GO TO 107
  126 IF (IA) 129,129,127
  127 RT = AA(IA)
      IA = IA-1
      IFLG = 1
C
C SCALAR MULTIPLICATION
C
      DO 128 J=1,M
         Y(J) = RT*Y(J)
  128 CONTINUE
  129 IF (IFLG) 130,130,107
  130 DO 131 J=1,M
         YY(J) = REAL(Y(J))
  131 CONTINUE
      RETURN
      END
```

CPRODP

```
      SUBROUTINE CPRODP (ND,BD,NM1,BM1,NM2,BM2,NA,AA,X,YY,M,A,B,C,D,U,
     1                   Y,ISGN)
C
C PRODP APPLIES A SEQUENCE OF MATRIX OPERATIONS TO THE VECTOR X AND
C STORES THE RESULT IN YY        PERIODIC BOUNDARY CONDITIONS
C AND   COMPLEX  CASE
C
C BD,BM1,BM2 ARE ARRAYS CONTAINING ROOTS OF CERTIAN B POLYNOMIALS
C ND,NM1,NM2 ARE THE LENGTHS OF THE ARRAYS BD,BM1,BM2 RESPECTIVELY
C AA   ARRAY CONTAINING SCALAR MULTIPLIERS OF THE VECTOR X
C NA IS THE LENGTH OF THE ARRAY AA
C X,YY THE MATRIX OPERATIONS ARE APPLIED TO X AND THE RESULT IS YY
C A,B,C  ARE ARRAYS WHICH CONTAIN THE TRIDIAGONAL MATRIX
C M  IS THE ORDER OF THE MATRIX
C D,U,Y ARE WORKING ARRAYS
C ISGN  DETERMINES WHETHER OR NOT A CHANGE IN SIGN IS MADE
C
      DIMENSION       A(1)      ,B(1)      ,C(1)      ,X(1)      ,
     1                Y(1)      ,D(1)      ,U(1)      ,BD(1)     ,
     2                BM1(1)    ,BM2(1)    ,AA(1)     ,YY(1)
      COMPLEX         Y         ,D         ,U         ,V         ,
     1                DEN       ,BM        ,YM        ,AM        ,
     2                Y1        ,Y2        ,YH        ,BD        ,
     3                CRT
      IF (ND) 102,102,101
  101 IF (ISGN) 104,104,102
  102 DO 103 J=1,M
         Y(J) = CMPLX(X(J),0.)
  103 CONTINUE
      GO TO 106
  104 DO 105 J=1,M
         Y(J) = CMPLX(-X(J),0.)
  105 CONTINUE
  106 MM = M-1
      MM2 = M-2
      ID = ND
      M1 = NM1
      M2 = NM2
      IA = NA
  107 IFLG = 0
      IF (ID) 114,114,108
  108 CRT = BD(ID)
      ID = ID-1
      IFLG = 1
C
C BEGIN SOLUTION TO SYSTEM
C
      BM = B(M)-CRT
      YM = Y(M)
      DEN = B(1)-CRT
      D(1) = C(1)/DEN
      U(1) = A(1)/DEN
      Y(1) = Y(1)/DEN
      V = CMPLX(C(M),0.)
      DO 109 J=2,MM2
         DEN = B(J)-CRT-A(J)*D(J-1)
```

```
CPRODP
                D(J) = C(J)/DEN
                U(J) = -A(J)*U(J-1)/DEN
                Y(J) = (Y(J)-A(J)*Y(J-1))/DEN
                BM = BM-V*U(J-1)
                YM = YM-V*Y(J-1)
                V = -V*D(J-1)
  109 CONTINUE
      DEN = B(M-1)-CRT-A(M-1)*D(M-2)
      D(M-1) = (C(M-1)-A(M-1)*U(M-2))/DEN
      Y(M-1) = (Y(M-1)-A(M-1)*Y(M-2))/DEN
      AM = A(M)-V*D(M-2)
      BM = BM-V*U(M-2)
      YM = YM-V*Y(M-2)
      DEN = BM-AM*D(M-1)
      IF (CABS(DEN)) 110,111,110
  110 Y(M) = (YM-AM*Y(M-1))/DEN
      GO TO 112
  111 Y(M) = (1.,0.)
  112 Y(M-1) = Y(M-1)-C(M-1)*Y(M)
      DO 113 J=2,MM
         K = M-J
         Y(K) = Y(K)-D(K)*Y(K+1)-U(K)*Y(M)
  113 CONTINUE
  114 IF (M1) 115,115,117
  115 IF (M2) 126,126,116
  116 RT = BM2(M2)
      M2 = M2-1
      GO TO 122
  117 IF (M2) 118,118,119
  118 RT = BM1(M1)
      M1 = M1-1
      GO TO 122
  119 IF (ABS(BM1(M1))-ABS(BM2(M2))) 121,121,120
  120 RT = BM1(M1)
      M1 = M1-1
      GO TO 122
  121 RT = BM2(M2)
      M2 = M2-1
C
C MATRIX MULTIPLICATION
C
  122 YM = Y(1)
      Y1 = (B(1)-RT)*Y(1)+C(1)*Y(2)+A(1)*Y(M)
      IF (MM-2) 125,123,123
  123 DO 124 J=2,MM
         Y2 = A(J)*Y(J-1)+(B(J)-RT)*Y(J)+C(J)*Y(J+1)
         Y(J-1) = Y1
         Y1 = Y2
  124 CONTINUE
  125 Y(M) = A(M)*Y(M-1)+(B(M)-RT)*Y(M)+C(M)*YM
      Y(M-1) = Y1
      IFLG = 1
      GO TO 107
  126 IF (IA) 129,129,127
  127 RT = AA(IA)
      IA = IA-1
```

```
CPROOP
      IFLG = 1
C
C SCALAR MULTIPLICATION
C
      DO 120 J=1,M
         Y(J) = RT*Y(J)
  120 CONTINUE
  129 IF (IFLG) 130,130,107
  130 DO 131 J=1,M
         YY(J) = REAL(Y(J))
  131 CONTINUE
      RETURN
      END
```

PPADD

```
      SUBROUTINE PPADD (N,IERROR,A,C,CBP,BP,BH,BN)
C
C PPADD COMPUTES THE ROOTS OF THE B SUP P POLYNOMIAL
C THIS ROUTINE IS CALLED AT THE LAST STEP OF THE PREPROCESSING PHASE
C IN THE CASE OF PERIODIC BOUNDARY CONDITIONS
C
C N IS THE ORDER OF THE BH AND BP POLYNOMIALS
C BP IS WHERE THE ROOTS OF THE B SUB P POLYNOMIAL ARE STORED
C CBP IS THE SAME AS BP EXCEPT TYPE COMPLEX
C BH IS USED TO TEMPORARILY STORE THE ROOTS OF THE B HAT POLYNOMIAL
C WHICH ENTERS THROUGH BP
C BN IS TEMPORARY STORAGE USED TO INDICATE THE TYPE OF ROOT IN BP
C WHETHER REAL, DOUBLE OR COMPLEX
C
      DIMENSION          A(1)        ,C(1)        ,BP(1)       ,BH(1)       ,
     1                   BN(1)       ,CBP(1)
      COMMON /CBLKT/     NPP         ,K           ,EPS         ,CNV         ,
     1                   L           ,NCMPLX      ,IK          ,IZ
      COMPLEX            CX          ,CSUM1       ,CSUM2       ,CPS0        ,
     1                   CHSG        ,CDD         ,CDC2        ,CPS0        ,
     2                   CFPPSG      ,CDX         ,CBP
      EXTERNAL           PSGF        ,PPSPF       ,PPSGF
      IZ = N
      IZM = IZ-1
      IZM2 = IZ-2
      DO 101 J=1,IZ
         BH(J) = BP(J)
  101 CONTINUE
      NCMPLX = 0
      XL = BH(1)
      DB = BH(3)-BH(1)
  102 XL = XL-DB
      IF (PSGF(XL,IZ,C,A,BH)) 102,103,103
  103 SGN = -1.
      CBP(1) = BSRH(XL,BH(1),IZ,C,A,BH,PSGF,SGN)
      XR = BH(IZ)
      DB = BH(IZ)-BH(IZ-2)
  104 XR = XR+DB
      IF (PSGF(XR,IZ,C,A,BH)) 104,105,105
  105 SGN = 1.
      CBP(IZ) = BSRH(BH(IZ),XR,IZ,C,A,BH,PSGF,SGN)
      DO 118 IG=2,IZM2,2
         XL = BH(IG)
         XR = BH(IG+1)
         SGN = -1.
         XM = BSRH(XL,XR,IZ,C,A,BH,PPSPF,SGN)
         PSG = PSGF(XM,IZ,C,A,BH)
         IF (ABS(PSG)-EPS) 108,108,106
  106    IF (PSG) 109,108,107
C
C     CASE OF A REAL ZERO
C
  107    SGN = 1.
         CBP(IG) = BSRH(BH(IG),XM,IZ,C,A,BH,PSGF,SGN)
         SGN = -1.
         CBP(IG+1) = BSRH(XM,BH(IG+1),IZ,C,A,BH,PSGF,SGN)
```

```
PPADD
            BN(IG) = 0.
            BN(IG+1) = 0.
            GO TO 118
C
C     CASE OF A MULTIPLE ZERO
C
  108       BN(IG) = -1.
            BN(IG+1) = -1.
            CBP(IG) = CMPLX(XM,0.)
            CBP(IG+1) = CMPLX(XM,0.)
            GO TO 118
C
C     CASE OF A COMPLEX ZERO
C
  109       PPSG = PPSGF(XM,IZ,C,A,BM)
            IF (PPSG) 110,121,110
  110       BHLD = SQRT(ABS(2.*PSG/PPSG))
            BN(IG) = 1.
            BN(IG+1) = -2.
            CBP(IG) = CMPLX(XM,BHLD)
            NCMPLX = 1
            CX = CBP(IG)
            IT = 0
  111       CSUM1 = (0.,0.)
            CFSG = (1.,0.)
            CMSG = (1.,0.)
            DO 112 J=1,IZ
               CDD = (CX-BM(J))
               CDD = 1./CDD
               CSUM1 = CSUM1+CDD
               CFSG = CFSG*A(J)*CDD
               CMSG = CMSG*C(J)*CDD
  112       CONTINUE
            CPSG = (1.,0.)-CFSG-CMSG
            IF (CABS(CPSG)) 121,117,113
  113       IGM = IG-1
            CSUM2 = 0.
            DO 114 J=1,IGM
               CSUM2 = CSUM2+1./(CX-CBP(J))
  114       CONTINUE
            CDX = CSUM2-CSUM1/CPSG
            IF (CABS(CDX)) 121,121,115
  115       CDX = 1./CDX
            CX = CX+CDX
            IT = IT+1
            IF (IT-50) 116,116,121
  116       IF (CABS(CDX)-CNV) 117,117,111
  117       CBP(IG) = CX
            CBP(IG+1) = CCNJG(CX)
  118  CONTINUE
       IF (NCMPLX) 122,119,122
  119  DO 120 J=2,IZ
          BP(J) = REAL(CBP(J))
  120  CONTINUE
       GO TO 122
  121  IERROR = 4
```

```
  122 CONTINUE
      RETURN
      END                                                    PRADD
```

```
PSGF
          FUNCTION PSGF (X,IZ,C,A,BM)
          DIMENSION       A(1)       ,C(1)       ,BM(1)
          FSG = 1.
          MSG = 1.
          DO 101 J=1,IZ
             DD = 1./(X-BM(J))
             FSG = FSG*A(J)*DD
             MSG = MSG*C(J)*DD
      101 CONTINUE
          PSGF = 1.-FSG-MSG
          RETURN
          END
```

BSRH

```
      FUNCTION BSRH (XLL,XRR,IZ,C,A,BH,F,SGN)
      DIMENSION        A(1)          ,C(1)          ,BH(1)
      COMMON /CBLKT/   NPP           ,K             ,EPS           ,CNV
     1                 L             ,NCMPLX        ,IK            ,IZ
      XL = XLL
      XR = XRR
  101 X = .5*(XL+XR)
      IF (SGN*F(X,IZ,C,A,BH)) 103,105,102
  102 XR = X
      GO TO 104
  103 XL = X
  104 IF (XR-XL-CNV) 105,105,101
  105 BSRH = .5*(XL+XR)
      RETURN
      END
```

PPSGF

```
      FUNCTION PPSGF (X,IZ,C,A,BH)
      DIMENSION       A(1)        ,C(1)        ,BH(1)
      SUM = 0.
      DO 101 J=1,IZ
         SUM = SUM-1./(X-BH(J))**2
  101 CONTINUE
      PPSGF = SUM
      RETURN
      END
```

PPSPF

```
      FUNCTION PPSPF (X,IZ,C,A,BN)
      DIMENSION       A(1)        ,C(1)        ,BN(1)
      SUM = 0.
      DO 101 J=1,IZ
         SUM = SUM+1./(X-BN(J))
  101 CONTINUE
      PPSPF = SUM
      RETURN
      END
```

COMPB

```
      SUBROUTINE COMPB (N,IERROR,AN,BN,CN,B,AH,BH)
C
C     COMPB COMPUTES THE ROOTS OF THE B POLYNOMIALS USING THE EISPACK
C     SUBROUTINE INTQL1.  IERROR IS SET TO 4 IF IF EITHER INTQL1 FAILS
C     OR IF A(J+1)*C(J) IS LESS THAN ZERO FOR SOME J.
C     AH,BH ARE TEMPORARY WORK ARRAYS
C
      DIMENSION       AN(1)      ,BN(1)      ,CN(1)      ,B(1)      ,
     1                AH(1)      ,BH(1)
      COMMON /CBLKT/  NPP        ,K          ,EPS        ,CNV       ,
     1                L          ,NCMPLX     ,IK         ,I2
      EPS = 1.
  101 EPS = EPS/10.
      DIF = 1.+EPS
      DIFM = DIF
      IF (DIFM-1.) 102,102,101
  102 EPS = 100.*EPS
      BNORM = ABS(BN(1))
      DO 103 J=2,N
         BNORM = AMAX1(BNORM,ABS(BN(J)))
  103 CONTINUE
      CNV = EPS*BNORM
      DO 110 J=1,N
         IF (J-1) 105,104,105
  104    IF (NPP) 110,105,110
  105    ARG = AN(J)*CN(J)
         IF (ARG) 106,107,107
  107    CHLD = SQRT(ARG)
         IF (CN(J)) 109,108,108
  108    B(J) = CHLD
         GO TO 110
  109    B(J) = -CHLD
  110 CONTINUE
      IF = 2**K
      LH = IF
      LH1 = IF+1
      I1 = 1
      DO 122 L=1,K
         I1 = I1+I1
         NN = I1+I1-1
         IFL = IF-I1+1
         DO 121 I=1,IFL,I1
            IF (I-IFL) 116,111,111
  111       IF (NPP) 112,113,112
  112       LH = LH+NN
            GO TO 121
  113       LS = 0
            DO 114 J=I,IF
               LS = LS+1
               BH(LS) = BN(J)
               AH(LS) = B(J)
  114       CONTINUE
            I1M = I1-1
            DO 115 J=1,I1M
               LS = LS+1
               BH(LS) = BN(J)
```

COMPB

```
             AM(LS) = B(J)
115        CONTINUE
           GO TO 118
116        LS = 0
           JFL = I+NN-1
           DO 117 J=I,JFL
             LS = LS+1
             BM(LS) = BN(J)
             AM(LS) = B(J)
117        CONTINUE
118        CALL TOLRAT (NN,BM,AM,IERROR)
           IF (IERROR) 106,119,106
119        DO 120 J=1,NN
             LM = LM+1
             B(LM) = -BM(J)
120        CONTINUE
121      CONTINUE
         LM1 = LM+1
122  CONTINUE
     DO 123 J=1,N
       B(J) = -BN(J)
123  CONTINUE
     IF (NPP) 128,124,128
124  J2 = 2*N*(K-1)+4
     LM = J2
     J1 = J2-N-N+1
     N2M2 = J2+N+N-4
125  D1 = ABS(B(J1)-B(J2-1))
     D2 = ABS(B(J1)-B(J2))
     D3 = ABS(B(J1)-B(J2+1))
     IF ((D2 .LT. D1) .AND. (D2 .LT. D3)) GO TO 126
     B(LM) = B(J2)
     J2 = J2+1
     LM = LM+1
     IF (J2-N2M2) 125,125,127
126  J2 = J2+1
     J1 = J1+1
     IF (J2-N2M2) 125,125,127
127  B(LM) = B(N2M2+1)
     J1 = 2*N*(K-1)+3
     J2 = J1+N+N+N
     J3 = J2+N
     CALL PPADD (N,IERROR,AN,CN,B(J1),B(J1),B(J2),B(J3))
     JF = J1+N-1
128  RETURN
106    IERROR = 4
     RETURN
     END
```

TOLRAT

```
      SUBROUTINE TOLRAT(N,D,E2,IERR)
C
      INTEGER I,J,L,M,N,II,L1,MML,IERR
      REAL D(N),E2(N)
      REAL B,C,F,G,H,P,R,S,MACHEP
C     REAL SQRT,ABS,SIGN
      COMMON /CBLKT/ NPP        ,K          ,MACHEP     ,CNV        ,
C
C     THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE TOLRAT.
C     ALGORITHM 464, COMM. ACM 16, 689(1973) BY REINSCH.
C
C     THIS SUBROUTINE FINDS THE EIGENVALUES OF A SYMMETRIC
C     TRIDIAGONAL MATRIX BY THE RATIONAL QL METHOD,
C
C     ON INPUT-
C
C        N IS THE ORDER OF THE MATRIX,
C
C        D CONTAINS THE DIAGONAL ELEMENTS OF THE INPUT MATRIX,
C
C        E2 CONTAINS THE                   SUBDIAGONAL ELEMENTS OF THE
C          INPUT MATRIX IN ITS LAST N-1 POSITIONS.  E2(1) IS ARBITRARY.
C
C      ON OUTPUT-
C
C        D CONTAINS THE EIGENVALUES IN ASCENDING ORDER.  IF AN
C          ERROR EXIT IS MADE, THE EIGENVALUES ARE CORRECT AND
C          ORDERED FOR INDICES 1,2,...IERR-1, BUT MAY NOT BE
C          THE SMALLEST EIGENVALUES,
C
C        E2 HAS BEEN DESTROYED,
C
C
C          ZERO       FOR NORMAL RETURN,
C          J          IF THE J-TH EIGENVALUE HAS NOT BEEN
C                     DETERMINED AFTER 30 ITERATIONS.
C
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
C     APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C
C
C     ********** MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING
C                THE RELATIVE PRECISION OF FLOATING POINT ARITHMETIC.
C
C                **********
      IERR = 0
      IF (N .EQ. 1) GO TO 1001
C
      DO 100 I = 2, N
  100 E2(I-1) = E2(I)*E2(I)
C
      F = 0.0
      B = 0.0
      E2(N) = 0.0
C
```

```
TOLRAT
          DO 290 L = 1, N
             J = 0
             H = MACHEP * (ABS(D(L)) + SQRT(E2(L)))
             IF (B .GT. H) GO TO 105
             B = H
             C = B * B
C     ********** LOOK FOR SMALL SQUARED SUB-DIAGONAL ELEMENT **********
  105     DO 110 M = L, N
             IF (E2(M) .LE. C) GO TO 120
C     ********** E2(N) IS ALWAYS ZERO, SO THERE IS NO EXIT
C                THROUGH THE BOTTOM OF THE LOOP **********
  110     CONTINUE
C
  120     IF (M .EQ. L) GO TO 210
  130     IF (J .EQ. 30) GO TO 1000

C     ********** FORM SHIFT **********
          L1 = L + 1
          S = SQRT(E2(L))
          G = D(L)
          P = (D(L1) - G) / (2.0 * S)
          R = SQRT(P*P+1.0)
          D(L) = S / (P + SIGN(R,P))
          H = G - D(L)
C
          DO 140 I = L1, N
  140     D(I) = D(I) - H
C
          F = F + H
C     ********** RATIONAL QL TRANSFORMATION **********
          G = D(M)
          IF (G .EQ. 0.0) G = B
          H = G
          S = 0.0
          MML = M - L
C     ********** FOR I=M-1 STEP -1 UNTIL L DO -- **********
          DO 200 II = 1, MML
             I = M - II
             P = G * H
             R = P + E2(I)
             E2(I+1) = S * R
             S = E2(I) / R
             D(I+1) = H + S * (H + D(I))
             G = D(I) - E2(I) / G
             IF (G .EQ. 0.0) G = B
             H = G * P / R
  200     CONTINUE
C
          E2(L) = S * G
          D(L) = H
C     ********** GUARD AGAINST UNDERFLOWED H **********
          IF (H .EQ. 0.0) GO TO 210
          IF (ABS(E2(L)) .LE. ABS(C/H)) GO TO 210

          IF (E2(L) .NE. 0.0) GO TO 130
  210     P = D(L) + F
```

```
TOLRAT
C     ********** ORDER EIGENVALUES **********
         IF (L .EQ. 1) GO TO 250
C     ********** FOR I=L STEP -1 UNTIL 2 DO -- **********
      ABSP=ABS(P)
         DO 230 II = 2, L
            I = L + 2 - II
            IF (ABSP .GE. ABS(D(I-1))) GO TO 270
            D(I) = D(I-1)
  230    CONTINUE
C
  250    I = 1
  270    D(I) = P
  290 CONTINUE
C
      GO TO 1001
C     ********** SET ERROR -- NO CONVERGENCE TO AN
C                EIGENVALUE AFTER 30 ITERATIONS **********
 1000 IERR = L
 1001 RETURN
C     ********** LAST CARD OF TOLRAT **********
      END
```